

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Rahel Pettai

**Kasutajaliidese disaini ja digiligipääsetavuse
edasiarendus „Kasutatavuse kogemuspõhise
analüüsi meetodi“ rakendusele**

Bakalaureusetöö (9 EAP)

Juhendaja:
Anu Piirisild, MSc

Tartu 2025

Kasutajaliidese disaini ja digiligipääsetavuse edasiarendus „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele

Lühikokkuvõte:

Bakalaureusetöö eesmärk on arendada edasi kasutatavuse kogemuspõhise analüüsi (EbA) meetodi töölaarakenduse kasutajaliidest ja eessüsteemi, keskendudes uute funktsionaalsuste arendamisele ning digiligipääsetavuse parandamisele vastavalt Web Content Accessibility Guidelines (WCAG) versiooni 2.2 suunistele ja Euroopa Liidu standardile EN 301 549. Töö sisaldab ülevaadet EbA meetodist ning digiligipääsetavuse põhimõtetest ja standarditest. Töös keskenduti kasutajaliidese funktsionaalsuste täiustamisele lähtuvalt digiligipääsetavuse aspektist, seal-hulgas klaviatuuriga navigeerimise ja ekraanilugeri toe loomisele. Töös kirjeldatakse loodud komponentide disainiprotsessi, nende implementeerimist ja testimist. Rakendus valmis neljaliikmelises tiimis koostöös tooteomanikuga, kus töö autori ülesandeks oli luua uutele funktsionaalsustele disainilahendus ning teha eessüsteemi arendustööd.

Võttesõnad:

Digiligipääsetavus, WCAG, Kasutatavuse kogemuspõhise analüüsi (EbA) meetod, eessüsteemi arendus, EN 301 549, kasutajaliidese disain, HTML, CSS, Typescript, Angular

CERCS: P175 Informaatika

Further Development of User Interface and Digital Accessibility for the “Experience-based Usability Analysis Method” Application

Abstract:

The aim of this bachelor’s thesis was to further develop the user interface and front-end of the “Experience-based Usability Analysis Method” (EbA) desktop application, focusing on the implementation of new functionalities and the improvement of digital accessibility. The development was guided by the Web Content Accessibility Guidelines (WCAG) version 2.2 and the European Union standard EN 301 549. The thesis provides an overview of the EbA method, accessibility principles and guidelines. The main focus was on enhancing the user interface with accessibility in mind, including keyboard navigation and screen reader support. The design process, implementation and testing of new components are described in detail. The application was supervised by the product owner and developed by a four-member team, with the author responsible for the design and front-end development.

Keywords:

Digital accessibility, WCAG, Experience-based Analysis (EbA) Method, front-end development, EN 301 549, user interface design, HTML, CSS, Typescript, Angular

CERCS: P175 Informatics

Sisukord

Sissejuhatus.....	5
1. Taust.....	7
1.1 EbA meetod	7
1.2 Digiligipääsetavus.....	9
1.2.1 Digiligipääsetavuse standard WCAG	12
1.2.2 Euroopa Liidu ligipääsetavuse tehniline standard EN 301 549	13
1.3 EbA-UIM rakenduse arenduse esimene etapp.....	14
1.4 EbA-UIM rakenduse arenduse teine etapp	15
1.4.1 Digiligipääsetavuse kasutajagrupi skoop.....	16
1.4.2 Teise etapi arendusvajadused.....	16
1.4.3 Valitud ligipääsetavuse nõuded	18
1.4.3 Teise etapi arendusmeeskond ja ülesannete jaotus	19
2. EbA-UIM rakenduse kasutajaliidese disaini edasiarendus	20
2.1 Digiligipääsetavuse kriteeriumite analüüs ja vastavuse hindamine.....	20
2.2 Visuaalse kujunduse lahendused.....	21
2.2.1 Värvide valik uutes komponentides.....	22
2.3 Klaviatuuriga navigeerimine.....	23
3. EbA-UIM rakenduse eessüsteemi edasiarendus	26
3.1 Kasutatud tehnoloogiad	26
3.2 Rakenduse teises etapis arendatavad komponendid	26
3.2.1 Nupud.....	27
3.2.2 Näidete prioritiseerimine	29
3.2.3 Klaviatuuriga navigeerimine.....	30
3.2.4 Ekraanilugeri võimaldamine.....	32
3.2.5 Teavitused ja märguanded	33
3.2.6 Arendusest välja jäänud nõuded	34

4. Tulemused.....	37
4.1 Testimine.....	37
4.1.1 Üldine kasutatavuse testimine.....	37
4.1.2 Klaviatuuriga navigeerimise testimine	38
4.1.3 Ligipääsetavuse manuaalne testimine.....	40
4.2 Rakenduse valminud tulemus	41
Kokkuvõte.....	43
Viidatud kirjandus.....	44
Lisad.....	48
I. WCAG 2.2 kriteeriumite valik.....	48
II. Ülesannete jaotus eessüsteemi arendajate vahel	54
III. Klaviatuuri kasutatavuse testimise sisend.....	56
IV. Testitud nõuded.....	58
V. Litsents.....	60

Sissejuhatus

Digiligipääsetavus tähendab digitoodete ja -teenuste arendamist viisil, mis tagab nende kasutatavuse kõikidele inimestele olenemata nende tehnilistest vahenditest, asukohast või individuaalsetest erivajadustest [1]. Digitaalsete ja -teenuste digiligipääsetavus on oluline arendajatele ja organisatsioonidele, kes soovivad vältida olukorda, kus erivajadustega inimesed jäävad nende digitaalsest tootest või teenustest kõrvale, sest puudub võrdne ligipääs [1]. Digiligipääsetavuse laialdasemat rakendamist takistab stereotüüpne arvamus, et see puudutab vaid väikest sihtrühma, kuid selle seisukoha tekkimist on mõjutanud sihtrühma vähene nähtavus, mis omakorda tuleneb ligipääsetavuse puudulikkusest [2]. Sellest lähtuvalt keskendub käesolev bakalaureusetöö kasutajaliidese disaini ja digiligipääsetavuse edasiarendusele EbA-UIM rakenduses, mis on mõeldud kasutatavuse kogemus põhise analüüsi (EbA) meetodi (ingl *Experience-based Analysis Method*) kasutamiseks.

EbA meetod on loodud digitoodete või -teenuste analüüsimiseks, et tuvastada kasutajate varasema käitumise ja kogemuse pinnalt võimalikud riskikohad, parendusvõimalused või täiendav analüüsivajadus [3, 4]. Selles töös on aluseks võetud EbA-UIM rakenduse esimese etapi arendustöö tulem ja Anu Piirisilla välja töötatud EbA meetod, mille kasutamiseks eelnimetatud rakendus on tehtud. Rakenduse kõige esimene prototüüp oli algselt Microsoft Exceli tabel [5], kuid soovist viia see üle kasutajasõbralikumale kujule, arendati EbA-UIM rakenduse esimene versioon [6, 7]. Esimeses versioonis olid loodud rakenduse baas ja põhifunktsionaalsused, kuid vajadus täienduste järele lõi eelduse teiseks arendusetapiks, mis on käesoleva bakalaureusetöö fookuses.

EbA-UIM rakenduse edasiarendus toimub neljaliikmelises arendustiimis. Käesoleva bakalaureusetöö eesmärk on edasiarendada EbA-UIM rakenduse kasutajaliidest ning suurendada digiligipääsetavust, millest mõlemad hõlmavad eessüsteemi (ingl *front-end*) arendust. Rakenduse täiendamine eeldab kasutajaliidese uute komponentide loomist, mida tehakse vastavalt veebi sisu digiligipääsetavuse suunistele Web Content Accessibility Guidelines (WCAG) ja Euroopa Liidu standardile EN 301 549.

Bakalaureusetöö koosneb neljast peatükist. Esimeses peatükis kirjeldatakse EbA meetodi tööühimõtteid ja selle töövahendit, digiligipääsetavuse olemust ja vaatluse all olevaid suuniseid WCAG ja EN 301 549. Lisaks kirjeldatakse rakenduse arenduse esimest ja teist etappi. Teine peatükk annab ülevaate digiligipääsetavuse kriteeriumite analüüsist, nõuetest

esimeses ja teises etapis, kirjeldab rakenduse lahenduste kujunemise visuaalset poolt ning selgitab klaviatuuriga navigeerimise põhimõtteid. Kolmas peatükk kirjeldab arendatud lahendust. Neljas peatükk kajastab testimisi ja annab ülevaate tulemustest.

1. Taust

Peatükis antakse ülevaade EbA meetodist ja selle töövahendust, selgitatakse digiligipääsetavuse tähendust. Ülevaade antakse EbA-UIM töövahendi 1. etapist ning käesoleva töö raames tehtavat 2. etapi arendusplaanist.

1.1 EbA meetod

Töö eesmärk on arendada edasi rakendust, mis on mõeldud EbA meetodi kasutamiseks. EbA meetodit on järgnevalt kirjeldatud tuginedes seda tutvustavale artiklile [3], esmase töövahendi kirjeldusele [4] ja EbA-UIM rakenduses olevale EbA meetodi kasutusjuhendile¹. EbA meetod (ingl *Experience-Based Analysis Method*), eesti keeles „Kasutatavuse kogemuspõhise analüüsi meetod“ võimaldab digitoote või e-teenuse arendusprotsessi varajases faasis saada aimu loodava toote või teenuse kasutatavuse määra kohta tuginedes kasutajate varasemale käitumisele ja harjumustele. EbA analüüsi abil saab enne arendustööde algust prognoosida, kas lahendus planeeritud viisil toimib ning hinnata, kas kavandatud lahenduses tuleb midagi muuta, täiendada või täiendavalt analüüsida riskide ennetamiseks. Tavapäraselt on tootearenduse oluliseks osaks lõppkasutajatega testimine, kuid testimine lõppkasutajatega igal ajahetkel võib osutada kokkuvõttes ressursimahukaks. Selleks, et selgitada välja võimalikke kasutatavusega seotud probleeme juba tootearenduse varajases analüüsi faasis, on võimalik kasutada kasutatavuse inspekteerimise meetodeid (UIM, *Usability Inspection Method*). EbA meetod kuulub UIM meetodite alla ning on mõeldud kasutamiseks nõuete analüüsi tegijatele, kuid soovi korral ka tooteomanikele, ärianalüütikutele jt.

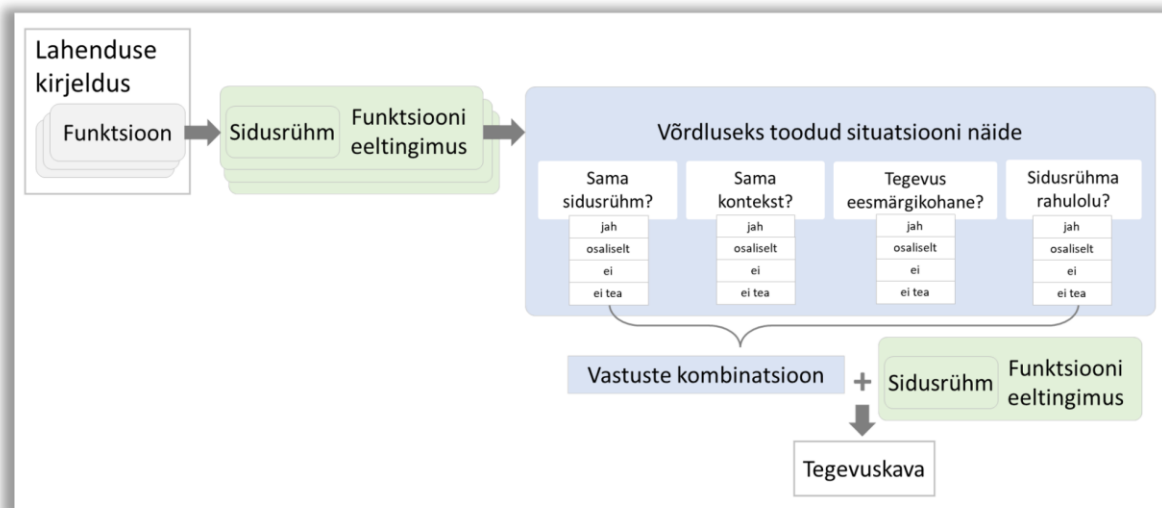
Järgnevas lõigus on kirjeldatud EbA meetodi rakendamise protsessi seda tutvustava artikli [3] ja EbA kasutusjuhendi¹ põhjal. Juhendi järgi koosneb meetod seitsmest etapist, mis on järgmised:

1. Kavandatud lahendus tuleb jagada väiksemateks iseseisvateks osadeks, näiteks funktsioonideks (ingl *features*). Funktsioonid on mõeldud lahenduse kirjeldamiseks, mis toovad välja väärtuspakkumine sidusrühmale ja aitavad kaasa toote eesmärgi saavutamisele.

¹ EbA-UIM rakendus: <https://bitbucket.org/eba-method/eba-code/downloads/>

2. Tuleb mõelda, millised on iga funktsiooniga seotud sidusrühmad (ingl *stakeholders*) ning ühes nendega ka funktsiooni edukaks toimimiseks vajalikud eeltingimused – tegevused, teadmised, oskused, hoiakud, vahendid ja ressursid.
3. Tuleb leida vastavalt eeltingimusele samastuv näide situatsioonist, mida sidusrühm on varasemalt juba kogenud. Näide ei pea olema tingimata vastava toote kohta, vaid võib olla ka mingis teises sarnases kontekstis. Sellisel juhul on oluline, et see siiski samastuks eeltingimusega.
4. Tuleb määrata, mil määral ühtib näide eeltingimusega ning hinnata sidusrühma tegevust toodud näites. See etapp koosneb neljast küsimusest, millest esimeses („Sama sidusrühm?“) tuleb hinnata kuivõrd ühtib näite ja eeltingimuse sidusrühm. Teises küsimuses („Sama kontekst?“) on vajalik määrata, kuivõrd vastab näites kirjeldatud kontekst eeltingimuses kirjeldatud kontekstile. Kolmas küsimus („Tegevus eesmärgikohane?“) palub hinnata, kas sidusrühm selles näite situatsioonis tegutses ettenähtud moel või omas neilt eeltatud oskuseid, teadmisi või ressursse. Viimane ehk neljas küsimus („Sidusrühma rahulolu?“) uurib sidusrühma rahulolu näitesituatsioonis temalt eeldatud tegevuse sooritamisel või selle kohta, et neil pidi olema mingi oskus, teadmine või ressursid.
5. Kui ühel eeltingimusel on mitu näidet, mille vastused neljandas etapis ei kattu, on soovitatav need vajadusel prioritseerida, et eemaldada fookusest kõrvalised näited. See aitab vähendada edasise analüüsi mahtu.
6. Neljandas etapis antud vastuste põhjal annab EbA meetodi töövahend automaatselt kombineeritud lause, mis annab analüüsijale indikatsiooni, kuivõrd vastab sidusrühma tegutsemine eeltingimuses nimetatud ootustele. Lause moodustub eelnevate vastuste kombinatsiooni tulemuse tõlgendusest ning varasemalt nimetatud sidusrühmast ja funktsiooni eeltingimusest. Kombinatsiooni tulemuse tõlgendusi on kokku 256.
7. Viimase etapina tuleb eelnevalt saadud vastuste põhjal teha järeldused ning soovi korral luua tegevuskava. Järelduste tegemiseks on oluline hinnata moodustunud kombineeritud lauseid, mis annavad aimu, kas eeltingimused on toote toimimiseks piisavad. Lahenduse paremaks toimimiseks võib näiteks planeerida nõuetes muudatusi, lisaanalüüsi või testimisi.

Eelpool kirjeldatud etapid on kokkuvõtlikult kajastatud joonisel 1, mis kujutab EbA meetodi keskseid elemente ja protsessi kulgu ühe funktsiooni näitel.



Joonis 1. EbA meetodi raamistik²

Enne EbA meetodi jaoks loodud esialgset rakendust oli selle esmaseks prototüübiks Microsoft Exceli tabel, mille kujundas Anu Piirisild oma magistritöö raames vastavalt MVP (ingl *Minimum Viable Product*) põhimõtetele [5]. Prototüüp sisaldas mitut lehte, mille seas oli andmete sisestamise vaade, kokkuvõtte lehekülg ning täiendavad juhendid mudeli tööloogika ja struktuuri mõistmiseks. Muud kasutajale suunatud juhised tabeli täitmise kohta olid lisatud Exceli paneelribale. Kaitstud magistritöös [5] loodud EbA meetodi töövahendi edasise arenduse ettepanekute seas tuuakse välja vajadus viia töövahend üle kasutajasõbralikumale platvormile. Sellele tuginedes planeeris EbA meetodi autor töövahendi arenduse mitmesse etappi. Käesoleva bakalaureusetöö autor liitus EbA meetodi rakenduse arendustiimiga hetkel, kui oli esimene arendusetapp juba teostatud ning alustati teise etapi ülesannete kaardistamisega. Esimeses etapis oli valminud rakenduse esimene versioon [6, 7], mida sai kasutada ning mis lõi eeldused rakenduse edasiseks arendamiseks kooskõlas digiligipääsetavuse suunistega. EbA meetodi rakenduse esimesest etapist antakse detailsem ülevaade peatükis 1.3.1.

1.2 Digiligipääsetavus

EbA meetodi rakenduse edasiarendusel on eesmärk keskenduda digitaalse ligipääsetavuse tõstmisele. Kui üleüldiselt määratletakse ligipääsetavust kui ümbritseva keskkonna ja selle elementide disainimist viisil, mis võimaldab nende kasutamist võimalikult suurele osale inimestele võrdsuse edendamiseks, siis digiligipääsetavus keskendub digitaalsetele

² EbA-UIM rakendus: <https://bitbucket.org/eba-method/eba-code/downloads/>

keskkondadele nagu veebilehed, rakendused ja e-teenused ning nende kättesaadavaks muutmisele kõikidele kasutajatele [8, 9] .

Ligipääsetavust käsitletakse elukaarepõhise lähenemisena, mille kohaselt võib igal inimesel elu jooksul tekkida püsiv või ajutine erivajadus, milleks võib olla tervislik seisund või ajutine vigastus [8]. Vananeva ühiskonna puhul nähakse suurt korrelatsiooni vananemise ja erivajaduste sagenemise vahel, sest vanusega suureneb tõenäosus kogeda erinevat liiki piiranguid, mis võivad mõjutada suutlikkust kasutada, tajuda ja mõista digitaalset sisu [10]. Seega ei ole see suunatud ainult kindlale sihtrühmale, vaid puudutab erinevas eluetapis suuremat osa ühiskonnast [8]. Ülemaailmselt elab hinnanguliselt 1,3 miljardit inimest, kellel on mingit vormi puue, moodustades ligikaudu 16% elanikkonnast [9]. Eestis kuulub 2024. aasta lõpu seisuga 116 432 inimest puuetega inimeste hulka, moodustades 8% rahvastikust [11].

Eesti keeles veebi- ja digilahenduste ligipääsetavuse käsitlemisel on kasutusel mõisted ligipääsetavus, juurdepääsetavus ja digiligipääsetavus. Antud töös kasutatakse läbivalt mõistet digiligipääsetavus, mida kasutatakse ka mitmetes Eesti dokumentides, näiteks Tarbijakaitse ja Tehnilise Järelevalve Ameti aastaaruannetes³, kus hinnatakse avaliku sektori veebi- ja mobiilirakenduste vastavust nõutud suunistele.

Kuigi ligipääsetavus on vajalik suurele osale ühiskonnast, on selle rakendamine paljudes keskkondades, eriti digitaalses ruumis, endiselt puudulik ning loob takistusi just neile, kes vajaksid kohandatud lahendusi kõige rohkem [12]. Selle üks põhjus on, et inimeste vajadused on erinevad ja sageli jäävad konkreetsete kasutajagruppide vajadused arendustes tähelepanuta [12].

Üheks digiligipääsetavate veebi- ja digilahenduste sihtgrupiks on piiratud nägemisvõimega kasutajad, kes võivad kogeda takistusi, mis ulatuvad hägusest nägemisest ja kontrastitundlikkusest kuni nägemisvõime täieliku puudumiseni ning tihti esineb ka probleeme ereduse, värvieristuse või ka vaatevälja ulatusega [13]. Nende kasutajate ligipääs digitaalsele sisule sõltub suuresti visuaalse liidese selgusest, kontrastsusest ja kohandamisvõimalustest, mida näitas ka Kiambati jt [14] uuring. Selle uuringu andmetel kogevad vaegnägijad inimesed

³ Eesti avaliku sektori digiligipääsetavuse aruanded: <https://ttja.ee/eraklient/tarbija-oigused/kaubandus-teenused/digiligipaasetavuse-tagamine#Aruanded>

digitaalsetes keskkondades sagedamini raskusi kui nägemispuudega kasutajad, kuna nende vajadused on mitmekesisemad ning ekraanilugejast üksi ei piisa.

Teiseks sihtrühmaks on ajutise füüsilise piiranguga või motoorse erivajadusega kasutajad, kellel võib esineda raskusi käte või sõrmede liigutamise, liigutuse täpsuse või kiirusega [15, 16]. Sellised erivajadused mõjutavad kasutajate võimet kasutada digisisu traditsiooniliste vahenditega, nagu hiir või klaviatuur. Ligipääsetavuse seisukohalt on oluline toetada alternatiivseid navigeerimisviise, nagu klaviatuuriga liikumine, hääljuhtimine või kohandatud sisendseadmed. Kui veebi- või digilahendus ei võimalda mugavat ja katkestuseta klaviatuuriga navigeerimist või nõuab tehnilisi hiireliigutusi, võib see kasutajakogemust rikkuda [15]. Tehniline hiireliigutus võib olla näiteks korruga hiire lohistamine ja nupu allhoidmine.

Lisaks nägemisvõimega ja mootorsetele takistustele puutuvad paljud kasutajad kokku ka tunnetuslike ja õppimisega seotud raskustega [16]. The World Wide Web Consortium (edaspidi W3C) toob välja, et näiteks keerukas keelekasutus, pikk tekst ilma visuaalse toeta või ebaloogiline struktuur võivad olla takistuseks kognitiivsete või vaimsete erivajadustega inimestele [16]. Ka animatsioonid, vilkuvad elemendid või keerulised mitmeetapilised vormid võivad osutada probleemseks [16].

Hoolimata teadmisest, et erinevatel kasutajagruppidel on spetsiifilised vajadused, viitab Seixas Pereira ja Duarte [12] uuring sellele, et ligipääsetavuse tagamine jääb arendusprotsessis sageli tahaplaanile juurdunud takistuste tõttu. Uuringust selgus, et digiligipääsetavuse rakendamist mõjutavad süsteemsed probleemid nagu ligipääsetavusalase hariduse ja koolituste puudumine. Sellest tulenevalt toodi välja ka teadlikkuse puudumine arendusprotsessis osalejate seas. Uuringus osalenud praktikud tõid esile ka ligipääsetavuse tagamisega kaasneva ajakulu ja rahalised väljaminekud. Näiteks on üheks oluliseks kitsaskohaks testimine – automatiseeritud tööriistad pole veel piisavalt pädevad, mistõttu tuleb paljudel juhtudel tugineda manuaalsele testimisele, mis on märkimisväärselt ajakulukam. Lisaks toodi välja, et olemasolevad ligipääsetavuse standardid võivad olla keerukad ja ebamäärase sõnastusega, mistõttu on raske neid praktikas rakendada.

Kesksemad probleemid, mis esinevad erinevates e-keskkondades ning puudutavad otseselt erinevaid kasutajagruppe, on analüüsinud ja kajastanud Eesti Vabariigi Valitsuse poolt kokku kutsutud ligipääsetavuse rakkerühm oma 2021. aasta lõpparuandes [2]. Selle kohaselt ei ole paljud avaliku sektori e-keskkonnad taganud vastavust nõutud digiligipääsetavuse nõuetega. Probleemidena on välja toodud muu hulgas riigi teenuste kehv leidmine otsingumootorist,

keerukas ja raskesti mõistetav keel, piiratud või puudulik tugi abitehnoloogiatele ning ebaühtlane struktuur lehtede vahel. Viimane raskendab orienteerumist ja nõuab kasutajalt täiendavat kohanemist, mille tulemusel võib tekkida õpiköver isegi lihtsamate toimingute sooritamiseks.

Rahvusvahelised juhised ja regulatsioonid, nagu Euroopa Liidu veebiligipääsetavuse direktiiv (ingl *Web Accessibility Directive*) ja standard EN 301 549, mida on kirjeldatud peatükis 1.2.2, on küll kehtestatud, kuid nende rakendamine ei ole ühtlane [12]. Avaliku sektori poolt pakutavad teenused peavad näiteks Euroopas Liidus järgima selleks ettenähtud digiligipääsetavuse standardit, kuid teistes sektorites on arusaam ligipääsetavusest vähesem ja selle rakendamine ei ole seadusandlikult reguleeritud ega kohustuslik [17, 12]. Rakendamise keerukus, standardite raskesti mõistetav sisu ning vähene toetus arendajatele on põhjused, miks digiligipääsetavus jääb praktikas tihti nõrgaks [12].

1.2.1 Digiligipääsetavuse standard WCAG

Järgnev lõik tutvustab digitaalse ligipääsetavuse standardit [18], mille kohaselt on Web Content Accessibility Guidelines (edaspidi WCAG) universaalne juhend. Juhendi on välja töötanud W3C mitmete sidusrühmade koostöös. Standardi eesmärk on parandada digitaalse keskkonna kasutatavust ja ligipääsetavust. WCAG-standardile on antud välja neli versiooni, mille käigus on järjepidevalt täiendatud standardi kolme vastavustaset - A, AA ja AAA. Kõige madalam on minimaalne kriteeriumite nõue ehk tase A, millele iga veebileht peaks vastama. Tase AA nõuab kõrgemat ligipääsetavust ning eeldab vastavust tasemele A. Kõrgeim tase on AAA, mis eeldab eelnevate tasemete nõuetele vastamist ning nõuab veelgi kõrgemaid rakendusmeetmeid ligipääsetavuse tõstmiseks, kuid selle saavutamine võib praktikas kujuneda keeruliseks või isegi võimatuks. Läbivalt on iga taseme suunised jaotatud järgmistesse kategooriatesse: tajutavus (ingl *perceivable*), talitlusvõime (ingl *operable*), mõistetavus (ingl *understandable*) ja töökindlus (ingl *robust*). Neid põhimõtteid tähistatakse ingliskeelse lühendiga POUR, mis moodustab WCAG standardi aluspõhja ning suunab arendajaid ja disainereid looma lahendusi, mis vastavad erinevate kasutajate vajadustele. Näiteks tajutavus hõlmab visuaalsete elementide loomisel kirjeldusi ja tekstialternatiive, kasutatavus aga tagab, et teenust saab kasutada ka ainult klaviatuuri või abiseadmete abil. Mõistetavus tähendab, et sisu ja funktsioonid peavad olema arusaadavad kõigile kasutajatele ning kohandatavus viitab sellele, et digisisu peab töötama abitehnoloogiatega ja erinevates seadmetes. POUR-raamistik

aitab seega hinnata, kas digitaalne sisu on ka praktiliselt kasutatav erinevate võimetega inimeste jaoks, lisaks sellele, et vastab tehnilistele nõuetele.

Kuigi digiligipääsetavuse tagamine pole kõikidel juhtudel seadusandlusega reguleeritud, siis rahvusvaheliselt on WCAG suunised kujunenud tunnustatud digiligipääsetavuse standardiks, mida rakendatakse paljude riikide õigusaktides või kohtupraktikas [18]. Näiteks Ameerika Ühendriikides nõuab föderaalsetele kehtiv ligipääsetavuse nõue Section 508, et veebisisu vastaks vähemalt WCAG 2.0 AA tasemele [18]. Samuti on käsitletav standard osa Euroopa Liidu standardist EN 301 549, mis on tehniline alus Euroopa Ligipääsetavuse Direktiivi (WAD) täitmiseks [18].

WCAG kriteeriumitele on vajalik teha ka hindamist, et kontrollida, kas vastavus kriteeriumitele on saavutatud. See hõlmab põhjalikku analüüsi toote funktsionaalsuse ja sisu üle, et tagada vastavus valitud kriteeriumitele. Kontrollimisse võib kuuluda näiteks värvide kontrastsustaseme hindamine, erinevate komponentide üles leidmine jms [19]. Seda protsessi saab läbi viia automaatselt erinevate tarkvarade abil, kuid mitte alati ei tuvasta need kõiki probleeme [19]. Sellega ei saa kindlaks teha sündmuste loogilist järgnevust, klaviatuuriga navigeerimist või tähenduslike alternatiivtekstide sisu, sest nende jaoks on vaja inimese hinnangut [20, 21]. Kuna automaattestimised ei tuvasta kõiki probleeme, on oluline teha ka manuaalne kontroll, mis hõlmab endas tehnika kasutatavuse, visuaalset ja sisulist läbivaatust [22]. WCAG puhul on selle jaoks vastava versiooniga kaasas ka täitmise tingimused ning kuidas neid saavutada [23, 24]. Nende kriteeriumite ja lahenduste järgimine ei pruugi tagada täielikku ligipääsetavust kõikidele kasutajatele, kuid aitab märkimisväärselt parandada digisisu ligipääsetavust nii erivajadustega inimestele kui ka teistele kasutajarühmadele, näiteks vanemaelastele, vaegnägijatele ja kognitiivsete erivajadustega kasutajatele [23].

1.2.2 Euroopa Liidu ligipääsetavuse tehniline standard EN 301 549

Käesolev lõik kirjeldab ja selgitab digiligipääsetavuse standardit EN 301 549 (V3.2.1) [25, 26], mis on Euroopa standard info- ja kommunikatsioonitehnoloogia (IKT) toodetele ja teenustele, et tagada digiligipääsetavus puuetega inimestele. Standard EN 301 549 on loodud Euroopa Liidu Direktiiv 2016/2102/EL kehtestatud miinimumnõuete rakendamiseks avaliku sektori veebilehtedele ja mobiilirakendustele. Lisaks sellele toetab standard ka Euroopa ligipääsetavuse akti (EAA), mis sarnaselt 2016/2102/EL direktiivile seab eesmärgiks tagada puuetega inimestele ligipääs info- ja kommunikatsioonitehnoloogia toodetele ja teenustele

kogu Euroopa Liidu ulatuses, kuid kohaldub ka erasektori teenustele nagu e-kaubandus, pangandus ja transporditeenused [27].

Käsitlev digiligipääsetavuse standard EN 301 549 tugineb suures osas WCAG 2.1 suunistele ja edukriteeriumitele, kuid hõlmab ka nõudeid, mis lähevad palju sügavamale, näiteks riistvarasüsteemide ligipääsetavuse suunised [28]. EN 301 549 on üles ehitatud nii, et seda oleks võimalik rakendada erinevate tehnoloogiliste toodete ja teenuste puhul [28]. Oma struktuurilt esitatakse standardis esmalt kasutajate vajadused funktsionaalsete kirjeldustena, mis selgitavad tehnoloogia kasutamiseks vajaminevat võimekust [26]. Seejärel on esitatud digiligipääsetavuse nõuded toote funktsioonide või omadustena. Lisaks sisaldab standard kahte olulist lisa: lisa A määratletakse, millised nõuded loetakse vastavaks Euroopa Liidu veebi ligipääsetavuse direktiivile 2016/2102/EL, ja lisa B kirjeldatakse, kuidas tehnilised nõuded aitavad täita kasutajate digiligipääsetavusega seotud vajadusi, mis on standardis sõnastatud funktsionaalsete eesmärkidena [26].

1.3 EbA-UIM rakenduse arenduse esimene etapp

EbA-UIM rakenduse esimese etapi arendus tehti kahe tudengi bakalaureusetöö raames. Rakenduse raamistiku ja eessüsteemi loogika kujundas vastavalt tooteomaniku nõuetele Karl Olaf Kuldmaa, mida on lähemalt kirjeldatud töös „Rakendusprogrammi loomine „Kasutatavuse kogemuspõhise analüüsi meetodile““ [7]. Rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded lõi koos visuaalse kujundusega Iris Kreinin (edaspidi esimese etapi disainer) oma bakalaureusetöö „Kasutajaliidese disaini loomine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele“ raames [6]. Lisaks teostas Iris Kreinin ka kasutajaliidese tehnilise teostuse vastavalt enda loodud disainile. Esimese etapi disaineri loodud kujundus, mis põhineb WCAG suunistel, ja selle tehniline teostus moodustavad aluse EbA-UIM rakenduse teises etapis läbiviidavatele eessüsteemi ja kasutajaliidese edasiarendusele, mis on käesoleva töö alguspunktiks.

Esimeses etapis valminud rakendus koosneb tagasüsteemist, eessüsteemist ja andmebaasist ning sisaldab mitut vaadet: projektide haldus, tabeli täitmine, kasutusjuhend ja rakendust tutvustav leht [7]. Tabeli vaates saab kasutaja lisada funktsioone, sidusrühmi ja võrreldavaid näiteid ning määrata hinnanguid, mille kombinatsiooni põhjal on genereeritud tooteomaniku poolt analüütilised vastused. Rakendus on loodud kasutamiseks nii eesti keeles kui ka inglise keeles [7].

Esimese etapi disainer määratles 43 funktsionaalset nõuet, mille eesmärk oli kirjeldada rakenduse jaoks vajalikke funktsionaalsusi. Need nõuded löid aluse disainilahenduse kujundamiseks, kuid osa nendest võeti aluseks ka teise etapi edasiarenduseks. Digiligipääsetavuse jaoks valiti esimeses etapis välja 16 kriteeriumi WCAG standardi 2.2 versioonist, mis kuuluvad tasemele A või AA ning väljendavad baastaseme prioriteetseid nõudeid ligipääsetava disaini loomiseks kasutajaliideses [6]. Valikusse kuulusid eelkõige disainikesksed suunised, mis puudutasid värvikasutust, kontraste, elementide märgistust ning navigeerimisloogikat. Valitud kriteeriumid ei hõlmanud tehnilist arendust ega otsest koodi valideerimist, vaid suunasid kujunduse funktsionaalset ja visuaalset planeerimist vastavalt WCAG põhimõtetele.

WCAG standardile vastavuse testimise viis Kreinin läbi kolmel tasandil – funktsionaalsus, kasutatavus ja ligipääsetavus, keskendudes eelkõige disainilahenduste vastavusele WCAG 2.2 välja valitud kriteeriumitele [6]. Funktsionaalsuse kontrollimiseks kasutati Given-When-Then loogikale tuginevat testimismudelit [32] ning ligipääsetavust hinnati WCAG kriteeriumite alusel visuaalse prototüübi ja rakenduse põhjal [6]. Kasutatavuse testimine tehti potentsiaalsete lõpp-kasutajatega, mille jaoks viidi läbi struktureeritud testimised. Kreinin on oma bakalaureusetöös välja toonud omapoolsed soovitused arendusvajaduste kohta rakenduse järgmistes arendusetappides, mis peamiselt sisaldavad loodud funktsionaalsuste täiendamist, kuid osalt ka WCAG kriteeriumite parema vastavuse saavutamist [6].

1.4 EbA-UIM rakenduse arenduse teine etapp

Esimesele etapile järgnes käesoleva bakalaureusetöö raames toimunud teine etapp, kus jätkati rakenduse edasiarendust ning fookus oli funktsionaalsuste laiendamisel. Teise etapi arendustöö tehti neljaliikmelises tiimis. Tiimitöö ülesannete jaotust on selgitatud peatükis 1.4.3. Käesoleva töö autor keskendus kasutajaliidese digiligipääsetavuse suurendamisele ja eessüsteemi edasiarendamisele ning seda kooskõlas ligipääsetavuse suunistega. Lähtepunkt sellele eesmärgile oli esimeses etapis valminud rakendus, sealt kaasa antud nõuded ning ka teise etapi käigus tekkinud uued nõuded.

Esimese etapi lahendused nõutud funktsionaalsustele lähtusid kasutajaliidese disaini suunitlusega välja valitud WCAG kriteeriumitest. Teise etapi alguses pärast nõuetega tutvumist, piiritleti koos tooteomanikuga kasutajagrupi skoop. Käesoleva töö autor analüüsis

teise etappi kavandatud arendusvajadusi lähtuvalt täpsustatud kasutajagruppide vajadusi rahuldavatele kriteeriumitele.

1.4.1 Digiligipääsetavuse kasutajagrupi skoop

EbA-UIM rakenduse tooteomaniku soov on, et lisaks tavakasutajale saaks rakendust kasutada ka piiratud nägemisvõimega inimesed ja mootorsete erivajadustega inimesed. Koos tooteomanikuga täpsustati selle põhjal rakenduse kasutajate skoop, mille alusel suurendati digiligipääsetavuse kriteeriumite valikut. Digiligipääsetavuse laiendamisel tuleb jälgida, et ka tavakasutajatel säiliks töövoogude kasutusmugavus, selgus ja efektiivsus.

Digiligipääsetavuse alla kuuluvatest sihtgruppidest valiti kasutajate skoopi ajutise füüsilise piirangu või motoorse erivajadusega inimesed ning ajutise või püsiva piiratud nägemisvõimega inimesed. Ajutise füüsilise piirangu või motoorse erivajadusega inimestele on vaja tagada klaviatuuriga navigeerimise võimalus. Kuna tooteomanik ei teadnud esimese etapi arendusnõuete komplekteerimisel klaviatuuriga navigeerimise vajadusele tähelepanu pöörata, tuli teise etapi arenduses üle vaadata, kas esimeses etapis realiseeritud lahendus võimaldab klaviatuuriga navigeerimist, vajadusel teha täiendusi ning uute funktsionaalsuste lisamisel jälgida, et need toetaksid klaviatuuriga navigeerimist. Ajutise või püsiva piiratud nägemisvõimega inimeste jaoks tuli lisanduvate disainielementide puhul säilitada värvide piisavat kontrastsustaset. Tagada tuli ka ekraanilugeri kohaldamise võimalus inimestele, kes seda vajavad.

1.4.2 Teise etapi arendusvajadused

Nõuded EbA-UIM rakenduse loomiseks koostas tooteomanik Anu Piirisild. Nagu peatükis 1.3 on välja toodud, täiendas esimese etapi disainer funktsionaalseid nõudeid tooteomaniku loodud sisendi põhjal. Nõuded prioritseeriti valides esimesse etappi välja need nõuded, mis toetavad rakenduse põhifunktsionaalsuseid. Teise etappi jäid rakenduse põhifunktsionaalsusi täiendavad ülesanded.

Käesoleva töö autori arenduskesksed ülesanded keskendusid eelkõige digiligipääsetavusele ning eeldasid sobiva disaini loomist. Mõningad ülesanded eeldasid ka parima lahenduse saamiseks pidevat info vahetust töö autori ja teise etapi arendustele analüüsi teinud nõuete

analüütiku vahel. Koostööd nõuete analüütikuga on täpsemalt on selgitatud peatükis 1.4.3. Töö autor liigitas enda ülesanded viite kategooriasse vastavalt keerukusastmele:

1. Väiksemad ja vähema ajakuluga ülesanded ja muudatused
 - a. “Kas + sidusrühm + eeltingimus” küsimuse moodustamise rõhutamine tabelivaates
 - b. Teises automaatselt liidetud lauses sidusrühma nimetuse rõhutamine
 - c. Projekti allalaadimise nupu disaini täiendamine
 - d. Projekti üles laadimise nupu disaini täiendamine
 - e. Jooksev vigade parandamine, mis olid tingitud uute funktsionaalsuste lisamisest
2. Keskmise raskusega funktsionaalsete nõuete täitmine
 - a. Tabeli kohal asuvate vahepaneelide peitmise ja avamise võimaluse loomine
 - b. Teise automaatselt liidetud lause tulba peitmise ja avamise võimaluse loomine
 - c. “Üles” nupu disaini ja funktsionaalsuse loomine
 - d. Nelikvastusele viienda variandi lisamine koos kaasneva funktsionaalsusega
 - e. Viimase tulba kategoriseerimise võimaluse loomine koos vastavate kirjeldustega
 - f. Veateadete hoiatussiltide implementeerimine
 - g. Keele eelistuse salvestamine kasutamise sessioonide vahel
3. Keskmise ja kuni kõrge keerukusastmega raskusega ülesanded
 - a. Prioritiseerimise võimaluse loomine
 - b. Prioritiseerimise tulemuse kuvamine
 - c. Ekraanilugeri võimaldamine
 - d. Klaviatuuril navigeerimise võimaldamine
4. Nõuded, mis vajasis esmalt analüüsi, et otsustada nende rakendamise võimalikkus
 - a. Tabeli vaates ridade järjekorra muutmine
 - b. Funktsionaalsuste lisamine ridade vahele
 - c. Nelikvastuse väljadele märkuste lisamine
 - d. Nupu “Lisa rida” asukoht

Ülesannete lahendamine ehk lahenduste loomine eeldas üldjuhul ka analüüsi. Leidus mõni ülesanne, mis vajab vaid ülesandele vastava muudatuse sisseviimist, sest analüüs oli juba eelnevalt tehtud. Ülejäänud lahendused sõltusid teises ja esimeses etapis välja valitud digiligipääsetavuse suunistest.

1.4.3 Valitud ligipääsetavuse nõuded

Rakenduse ligipääsetavuse suurendamiseks kooskõlas eesmärkide ja ülesannetega tuli vastu võtta uued ligipääsetavuse kriteeriumid, millele toetuda. Kui esimeses etapis keskenduti WCAG 2.2 versiooni A ja AA taseme välja valitud nõuetele, siis selles etapis laiendatakse WCAG 2.2 versiooni kriteeriumite valikut osaliselt AAA tasemeni. Mõnede ülesannete puhul arvestati ka nende Euroopa Liidu standardi EN 301 549 nõuetega, mis ei põhine WCAG-le.

Rakenduse teise etapi lahenduste loomise eeskujuks lisati esimeses etapis valitud kriteeriumitele täiendavalt järgmised kriteeriumid:

1. WCAG kriteerium 1.1.1 - „Mittetekstilise sisu”
2. WCAG kriteerium 1.3.1 - „Teave ja seosed”
3. WCAG kriteerium 1.3.2 - „Tähtsuse järjekord”
4. WCAG kriteerium 1.3.6 - „Eesmärgi tuvastamine”
5. WCAG kriteerium 1.4.13 - „Sisu lohistamisel või fookuses”
6. WCAG kriteerium 2.1.1 - „Klaviatuur”
7. WCAG kriteerium 2.1.2 - „Klaviatuurilõksu puudumine”
8. WCAG kriteerium 2.1.3 - „Klaviatuur (eranditena)”
9. WCAG kriteerium 2.4.3 - „Fokuseerimise järjekord”
10. WCAG kriteerium 2.4.7 - „Nähtav fookus”
11. WCAG kriteerium 2.4.8 - „Asukoht”
12. WCAG kriteerium 2.4.9 - „Lingi eesmärk”
13. WCAG kriteerium 2.4.11 - „Fookus pole varjatud (minimaalne)”
14. WCAG kriteerium 2.4.12 - „Fookus pole varjatud (täiustatud)”
15. WCAG kriteerium 2.4.13 - „Fookuse välimus”
16. WCAG kriteerium 2.5.2 - „Osutaja tühistamine”
17. WCAG kriteerium 3.1.1 - „Lehekülje keel”
18. WCAG kriteerium 3.2.1 - „Fookus”
19. WCAG kriteerium 3.2.5 - „Päringupõhine muutus”
20. WCAG kriteerium 3.3.5 - „Abi”
21. WCAG kriteerium 3.3.6 - „Vigade ennetamine (kõik)”
22. WCAG kriteerium 4.1.2 - „Nimi, roll, väärtus”
23. WCAG kriteerium 4.1.3 - „Oleku sõnumid”

Lisaks nendele kriteeriumitele tuleb tagada vastavus ka esimese etapi disaineri testitud kriteeriumitele. WCAG kriteeriumite valikud esimeses ja teises etapis tabelina koos valikute põhjendustega on välja toodud lisas 1.

Euroopa Liidu standardi EN 301 549 nõuetest kohalduvad EbA-UIIM rakenduse puhul järgmised kriteeriumid:

1. EN 301 549 kriteerium EN 5.2 - „Ligipääsetavusfunktsioonide aktiveerimine”
2. EN 301 549 kriteerium EN 11.6.1 - „Kasutaja kontroll ligipääsetavusfunktsioonide üle”
3. EN 301 549 kriteerium EN 11.6.2 - „Ligipääsetavusfunktsioonide mitte häirimine”
4. EN 301 549 kriteerium EN 12.1.1 - „Ligipääsetavuse ja ühilduvuse funktsioonid”

1.4.3 Teise etapi arendusmeeskond ja ülesannete jaotus

EbA meetodi jaoks loodud rakenduse edasiarendamiseks moodustati neljaliikmeline tiim. Igal liikmel oli eraldiseisev eesmärk ning sellele vastavad individuaalsed ülesanded. Siiski jagati ühte laiemat eesmärki – rakenduse edasiarendus ja väärtuse tõstmine, mille täide viimine eeldas meeskonnatööd. Töö meeskonnas eeldab tiimi juhtimist. Tiimijuhi ülesanded võttis enda kanda Sander Põldmaa (edaspidi tiimijuht) oma bakalaureusetöö „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakenduse edasiarendus ja tiimitöö juhtimine” raames [29]. Tiimijuht viis läbi iganädalasi koosolekuid, jälgis tööülesannete kulgu ja pakkus eessüsteemi arendajatele tehnilist tuge. Lisaks sellele tegeles tiimijuht ka taga- ja eessüsteemi edasiarenduse ning rakenduse üle viimisega Electronile (edaspidi kompleksarendaja). Gerdo Germanni (edaspidi tarkvara testija) roll tiimitöös oli testide koostamine ja läbiviimine enda bakalaureusetöö „Testimised ja macOSile kohandamine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele” raames [30]. Testimise alla kuuluvad käsitestimised, automatiseeritud testid ja kasutatavuse testimine. Germann töötas samuti rakenduse Electronile üleviimisega, kuid teostas ka edasiarenduse kohandamist macOS’ile ja Linuxil. Kolmandaks vajas arendustiim ka nõuete analüütikut. Selleks oli Timo Kaasik (edaspidi nõuete analüütik) ning tema ülesanne oli analüüsida nõudeid, pakkuda lahendusi ning osa neist ka rakendada eessüsteemis. Käesoleva töö autori ülesanded ja eesmärgid olid seotud eessüsteemi ja kasutajaliidese disaini edasiarendusega. See eeldas ka digiligipääsetavuse aspektist nõuetele analüüsi tegemist, sellele vastava disaini loomist ning seejärel kontrollimist. Teise etapi tööde arendusse oli kaasatud ka Anu Piirisild, kes tooteomanikuna jälgis, et arendus kulgeks kooskõlas tema visiooniga ning vastaks analüüsi käigus määratletud nõuetele ja eesmärkidele.

Tiimitööks toimusid suhtlusplatvormi Zoom vahendusel iganädalased koosolekud, kus räägiti iga meeskonnaliikme töö käigust – mis ülesannetega on nädala jooksul tööd tehtud, mis on valmistanud probleeme või millised osad nõuavad oodatust rohkem aega. Nädala pikkused töötsükliid ehk sprindid määrasid kohtumiste perioodilisuse. Teise etapi alguses jagati igale tiimiliikmele sõltuvalt eesmärgist ülesanded ning iga sprindi alguses määrati, milliste ülesannetega sellel konkreetsel nädalal tegeletakse. Ülevaate hoidmiseks kasutati tiimijuhhi poolt üles seatud Jira keskkonda. Kõik ülesanded olid alguses kogutud tööde nimekirja (ingl *backlog*) ning sealt edasi liigutati need vastavalt sellele, mis etapis kellegi ülesanded olid: “vaja teha”, “arendus”, “*pull request*”, “*testing*”, “*tested*”, “tehtud”.

2. EbA-UIM rakenduse kasutajaliidese disaini edasiarendus

Siin peatükis selgitatakse EbA-UIM rakenduse kasutajaliidese edasiarendamise protsessi ning kirjeldatakse klaviatuuriga navigeerimise põhimõtteid.

2.1 Digiligipääsetavuse kriteeriumite analüüs ja vastavuse hindamine

EbA-UIM rakenduse kasutajaliidese üheks oluliseks alguspunktiks oli digiligipääsetavuse nõuete järgimine. Töö teises etapis kontrolliti esimese arendusetapi vastavust ning keskenduti kasutajagrupi vajadustele lähtuvalt WCAG 2.2 ja EN 301 549 standardist. Ligipääsetavuse hindamine esimeses etapis tehtule toimus eeskätt Figma disainitööriista, visuaalse kontrolli ja WCAG vastava suunise edukriteeriumi alusel. Eesmärk oli hinnata ja tuvastada võimalikke parandusi vajavaid kohti. Analüüsi käigus tulid välja katmata kriteeriumid, mis olid välja toodud ka esimese etapi disaineri poolt.

WCAG kriteerium 2.4.5 ehk „Mitu võimalust” ei olnud esimeses etapis täidetud menüü navigeerimisvõimaluste vähesuse tõttu. Nimetatud kriteerium nõuab, et oleks tagatud mitu võimalust ühele lehele jõudmiseks, kuid menüüreal olevatele lehtedele „EbA meetodi juhend” ja „Meist” saab vaid menüüs olevatelt linkidelt. Esimese etapi disaineri töös oli kirjeldatud, et avavaade ehk „Projektid” leht vastab kriteeriumile, sest disaini põhjal on sinna võimalik navigeerida menüüst nii “Projektid” lingilt kui ka toote nime („EbA-UIM”) kaudu, kuid toote nime kaudu navigeerimise teostamine jäi teise etappi. Kriteeriumi 3.3.3 ehk „Soovitus vea korral” puudulikkus oli teadlikult fikseeritud ning ei kuulunud esimese etapi realiseerimisse, vaid järgmise etapi ülesannete hulka.

Lisaks varasema arendusetapi hindamisele viis töö autor teises etapis läbi ka uute funktsionaalsuste ligipääsetavuse analüüsi. Iga ülesande puhul hinnati vastavust WCAG 2.2 edukriteeriumitele ja vajadusel ka EN 301 549 standardis sätestatud tehnilistele nõuetele. Analüüs põhines funktsionaalsuste kirjeldustel, disainilahendustel ja kasutajaskonna vajadustel, mille puhul iga funktsionaalsuse puhul seoti vastavad kriteeriumid konkreetsete kasutusjuhtude ja ligipääsetavuse probleemidega. Analüüsi tulemusena koostati lahenduste kavandid, mis toetavad paremat juurdepääsu nii motoorse erivajadusega kui ka muude erivajadustega kasutajatele, kes vajavad värvide kontrastsuseid, erinevaid navigeerimisvõimalusi või ka ekraanilugeri tuge.

2.2 Visuaalse kujunduse lahendused

Esimeses etapis loodud EbA-UIM rakenduse kasutajaliidese kujundus baseerub visuaalselt rahulikult tonaalsusel, kasutades peamiste taustavärvidena rohelist ja beeži. Need toonid loovad taotluslikult esimese etapi disaineri määratlusele vastavalt mugava ja rahuliku keskkonna [6]. Loodud komponentide puhul kasutati ümaraid ääri ja tugeva kontrastsusega toonide asemel heledamaid, mis samuti rõhutavad rahulikkust. Teises etapis lisandusid rakendusse mitmed uued komponendid, millele oma üldise kuju poolest sai edasi anda esimese etapi komponentidele sarnased omadused, kuid värvilahendusi tuli laiendada viisil, mis säilitaks olemasoleva kujundusstiili ühtsuse ning toetaks samal ajal kasutajakogemust ning värvidele vastavaid ligipääsetavuse nõudeid.

Selle töö raames ja eesmärkide täitmiseks kasutati Figma⁴, et pakkuda lahendustele visuaalset tuge ja luua elementide prototüübid. Kuna rakenduse töötav kasutajaliides ja disainisüsteem oli esimeses etapis loodud, ei olnud vajadust luua suuremahulise interaktiivsusega prototüüpe. Selle asemel kasutati Figma lihtsate ja konkreetsete visuaalide loomiseks, mille eesmärk oli rõhutada kindlaid kasutajaliidese elemente.

Töö autori jaoks oli lahenduste loomisel olulisel kohal suhtlus nõuete analüütikuga, sest mõne nõude rakendamiseks oli vajalik analüütiku eelnev põhjalikum analüüs lähtuvalt tooteomaniku visioonist ja soovidest. Sellisel juhul järgnes sellele lahenduse leidmine, mis tuli uuesti läbi arutada nõuete analüütikuga, et kas on tegu vastava lahendusega. Osad nõuded ei vajanud ka taolist suuremat analüüsi ning need tuli lihtsalt teostada. Seejärel vastavalt kokkuleppele

⁴ Figma: <https://www.figma.com/about/>

selgitati välja nõude lahendus ning kui miski vajas edasist analüüsi või korrigeerimist, saadeti see tagasi nõuete analüütikule. Lisas 2 on toodud tabel, mis näitab nõuete täitmise voogu alates analüüsist kuni arenduseni.

2.2.1 Värvide valik uutes komponentides

Uutest komponentidest oli vaja luua mitu erinevate funktsionaalsustega nuppu. Esmalt loodi “Üles” nupp, mille eesmärk oli kasutajale tekitada kiirtee lehe alt tagasi üles, kui tabelisse luuakse analüüsi käigus palju ridu. Kuna “Üles” nupp liikus mõlemal taustavärvil, siis katsetati erinevaid värvi võimalusi, et tagada WCAG kriteeriumite 1.4.3 ja 1.4.11 kontrastsuse nõudele vastavus nii teksti ja teksti tausta kui ka nupu ja selle tausta vahel. Kuigi kontrastsuse nõude täitmiseks on piisav tagada näiteks nupu puhul vaid kontrastne piirjoon, siis selle lahenduse puhul otsustati tumedama ja tugevama kontrastsusega nupu kasuks, et tagada selle parem silmapaistvus ja samaaegne kontrast nii rohelise ja beeži suhtes. Samuti oli oluline, et nupp paistaks välja tabeli eri värvi komponentide seast.

Veeru peitmiseks mõeldud nupp “Peida veerg”, tabeli kohal asetsevate paneelide peitmiseks “Peida” nupp ja tegevusplaani kategooriate määramiseks kasutatav “Kategoriseeri tegevusplaan” nupp kujundati kontrastsuse nõuet 1.4.11 silmas pidades, kasutades lahendusena tumedat piirjoont. Esimesel kahel juhul tagati märgatavus läbi erksama helerohelise tooni, mis eristus taustast, kuid jäi visuaalselt kooskõlla rahuliku kujundusstiiliga. Kuna “Peida veerg” nupp asub veeru pealkirja sees, oli oluline leida tasakaal nähtavuse ja üldisesse stiili sulandumise vahel. “Kategoriseeri tegevusplaan” nupu puhul, mille vajutamisel kuvatakse värvikoodiga kategooriad, eelistati seevastu tumedamat tooni beežist, et tagada eristuvus värvikoodiga värvitud taustast ja tabeli beeži tausta vahel. Kõik kolm nuppu tuginevad küll samale kontrastsuse põhimõttele, kuid nende värvitoonide valik on kohandatud konkreetse kasutuskoha ja funktsionaalsuse järgi.

Tarvis oli ka kujundada teavituskavad, millest üks informeerib tekkinud võimalusest prioritseerida tabelis olevaid näiteid ning teine kuvab teavituse veeru “Tee järeldused ja loo tegevuskava” lahtritesse olukorras, kui sinna väljale on midagi juba kirjutatud, kuid pärast seda on sellele eelnevaid vastuseid muudetud. Prioritiseerimise teavituse puhul lähtuti vajadusest eristada erinevat tüüpi informatiivseid paneele. Esimese etapi disainer oli kujundanud neutraalsed beežid teavitused olukordadeks, kus kasutajalt küsitakse kinnitust mõne tegevuse kohta, näiteks kustutamine. Teises etapis lisandus uus teavitustüüp, mille eesmärk on

informeerida kasutajat uue funktsionaalsuse olemasolust. Selleks kasutati sinist värvi, et eristada seda varasematest neutraalsetest teavitustest ning rõhutada teavituse informatiivset ja mitte-kriitilist iseloomu. Sinise värvi kasutamine tugineb ka varasematele värvipsühholoogia käsitlustele, mille kohaselt lisaks rohelisele loob ka sinine rahustava ja stabiilse fooni [33]. Teade, mis kuvatakse veeru “Tee järeldused ja loo tegevuskava” lahtritesse, kujundati visuaalselt vastavalt esimese etapi disaineri disainile ning teostati rakenduses selle põhjal.

Rakendusse lisandus ka funktsionaalsus prioritseerida näiteid, mis võimaldas vähem olulised näited jätta analüüsi käigus kõrvale. Selleks sai prioriteetsed näited märgistada märkeruuduga ning mitte-prioriteetsete näidetega read said halli taustavärvi, mis aitas neid visuaalselt taandada, säilitades samas loetavuse ja toetades kasutaja fookust olulisel sisul.

2.3 Klaviatuuriga navigeerimine

Ligipääsetava kasutajaliidese loomisel on oluline arvestada kasutajatega, kes ei kasuta hiirt ega puuteplaati, vaid navigeerivad rakenduses klaviatuuri abil. See hõlmab nii ajutise piiranguga nagu näiteks käeluumurd kui ka mootorika häiretega inimesi, kelle sihtgruppi on rohkem selgitatud peatükis 1.4.1. Klaviatuuriga navigeerimiseks kasutatakse kindlaid mustreid ning on oluline neid jälgida, et vältida segadust klaviatuuriga navigeerija kasutajakogemuses. Neid mustreid on kirjeldatud tehniliste juhiste kogumis Accessible Rich Internet Applications Suite (edaspidi WAI-ARIA)⁵, mis on mõeldud digisisu ligipääsetavamaks muutmiseks erivajadustega inimestele .

WAI-ARIA klaviatuuri võimaluste arendamise juhiste⁶ põhjal on kõige olulisem tagada, et kõik funktsionaalsed ehk interaktiivsed elemendid nagu lingid, nupud, sisestusväljad ja valikud oleks navigeeritavad tabeldusklahvi (ingl *tab key*) abil edasi ning tõstuklahvi (ingl *shift key*) ja tabeldusklahvi kombinatsiooni abil tagasi. Edasi või tagasi liikudes peab ka visuaalne fookus liikuma vastavalt ning võimalusel otse interaktiivsele elemendile, välja arvatud juhul, kui mõni muu komponent on sobivam. Näiteks võib fookus liikuda järgmise interaktiivse elemendi asemel esmalt pealkirjade või teavituste peale, mis pole interaktiivsed elemendid. Seetõttu ei peaks ka fookus nendele liikuma, kuid olukorras kus need võivad sisaldada olulist infot, võib arendaja spetsiaalselt fookuse sellele suunata, et kasutajat teavitada enne edasisi tegevusi.

⁵ WAI-ARIA: <https://www.w3.org/WAI/standards-guidelines/aria/>

⁶ WAI-ARIA. Developing a Keyboard Interface: <https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface/>

WCAG kriteerium 2.1.1⁷ eeldab ka vastavuse saavutamiseks, et kui elementidel on aktiveerimise funktsionaalsus, peab seda saama teha sisestusklahvi (ingl *enter key*) või tühikuklahvi (ingl *space key*) abil.

Kasutajaliideses võib esineda ka menüüsid, mille klaviatuuriga navigeerimiseks on WAI-ARIA defineerinud nooleklahvipõhised mustrid⁸. Menüüd avanevad vertikaalselt või horisontaalselt. Esimesel juhul tuleb menüü aktiveerimine ehk avamine võimaldada sisestus- või tühikuklahvi abil ning tagada menüü valikute vahel liikumine üles- ja allanooleklahvidega. Kui tegu on horisontaalselt avaneva menüüga, siis avamine toimub nagu vertikaalselt avaneva menüü puhul, kuid menüü valikute vahel liikumise peab implementeerima nii, et seda saaks teha vasak-noole ja parem-noole klahviga. Kasutajale peab võimaldama menüü sulgemise väljumisklahviga (ingl *escape key*).

Järgnev lõik toob välja WAI-ARIA selgituste⁹ põhjal ülehüppamislinkide olulisuse. Digiligipääsetavas rakenduses tuleb pakkuda võimalust jätta vahele korduv sisu, nagu näiteks linkidega menüüriba või otsing. Selleks on hea tava kasutada ülehüppamislinke (ingl *skip link*), mida tuleb kuvada ja fookuseerida esimese asjana lehel. On ka juhud, kus need pole visuaalselt kuvatud enne klaviatuuri kasutamist, kuid sellisel juhul peavad need olema hästi märgatavad. Korduva info vahele jätmiseks mõeldud lingid pole kasulikud ainult klaviatuuri kasutajatele, vaid ka ekraanilugeri kasutajatele ja teistele inimestele, kellel võib esineda mootorikäireid või kasutada abivahendeid klaviatuuri kasutamiseks.

Disainilt on oluline klaviatuuriga navigeerimisel kindlaks teha, et fookuses olev element on teistest eristatav ning seda on kirjeldanud oma lehel ka WAI-ARIA⁶. Klaviatuuri kasutaja peab saama jälgida, kuidas fookus liigub, et anda edasi sarnane võimalus, nagu on hiirega kasutajal, kes jälgib kursorit. Hea eristuse tagamine on vajalik, et klaviatuuriga navigeerija ei kaotaks ära järge, kus ta lehel parasjagu asub. Osadel veebilehtedel ja rakendustel on vaikimisi määratud visuaalne eristus, mida on soovitatud kasutada, kuid see pole kohustuslik. Kohandatud lahenduse puhul on oluline arvesse võtta, et fookus ei kaoks kõrge kontrastsusega režiimis ja saaks teha vahet fookuse ja muu ära valitud elemendi vahel, mis võib esile olla tõstetud.

⁷ W3C. WCAG 2.2 Keyboard (Level A): <https://www.w3.org/WAI/WCAG22/Understanding/keyboard.html>

⁸ WAI-ARIA. Menu and Menubar Pattern: <https://www.w3.org/WAI/ARIA/apg/patterns/menubar/>

⁹ WAI-ARIA. Skip Link - Easy Checks: <https://www.w3.org/WAI/test-evaluate/easy-checks/skip-link/>

Vahekaartide olemasolul on oluline järgida samuti WAI-ARIA seatud navigeerimismustreid¹⁰. Vahekaardid on lehtede rühm, mis kuvab fookuse korral ühte lehte korraga. Klaviatuuriga navigeerija peab saama vahelehe rühmale fookuse suunata tabeldusklahviga, kuid ei saa liikuda sellega antud rühma lehtede vahel. Kui tegu on näiteks kahetasemelise vahelehe süsteemiga, saab tabeldusklahviga liikuda kahe taseme vahel, kuid mitte ühe vahekaardi rühma lehtede vahel nagu eelnevalt mainitud. Vahekaardi rühma lehtede vahel liikumiseks, kui need on horisontaalselt paigutatud, saab seda teha vasak-noole ja parem-noole klahviga. Sellisel juhul on üles-noole ja alla-noole klahv lehe kerimiseks. Kui vahelehed on vertikaalselt paigutatud, toimub selle rühma lehtede vahel liikumine üles-noole ja alla-noole klahviga. Vahekaardi avamiseks tuleb kasutada sisestus- või tühikuklahvi. Soovi korral saab kasutada ka *home* nuppu, mis viib vastava vahekaardi rühma lehtede seas esimesele lehele ning *end* nupp viib vahekaardi rühma lehtede viimasele lehele.

Viimasena selgitatakse klaviatuuriga navigeerimisel kehtivaid WAI-ARIA märkeruudu kasutamise mustreid¹¹. Fookuse saab sellele suunata teistele elementidele sarnaselt tabeldusklahvi abil. Selleks, et märkeruudu märget muuta, tuleb kasutada tühikuklahvi.

¹⁰ WAI-ARIA. Tabs Pattern: <https://www.w3.org/WAI/ARIA/apg/patterns/tabs/>

¹¹ WAI-ARIA. Checkbox Pattern: <https://www.w3.org/WAI/ARIA/apg/patterns/checkbox/>

3. EbA-UIM rakenduse eessüsteemi edasiarendus

Selles peatükis nimetatakse rakenduse arendamisel kasutatud tehnoloogiaid ning antakse ülevaade teises etapis loodud komponentidest ja arendusest välja jäänud nõuetest.

3.1 Kasutatud tehnoloogiad

Esimeses etapis valminud esialgsele rakendusele valis tehnoloogiad esimese etapi arendustiim, millest peeti kinni ka teise etapi jooksul. Eessüsteemis on kasutusel Angular raamistik, mis oma komponendipõhise struktuuri poolest hõlmab ka HTML¹², CSS¹³ kui ka TypeScript¹⁴. Tagasüsteem on üles ehitatud Spring raamistikul¹⁵ koos Java¹⁶, SQLite¹⁷ ja Maveniga¹⁸. Kui eelnevalt kasutati rakenduse kokku panemiseks Javat ja Launch4j¹⁹ installerit, siis teises etapis asendasid kompleksarendaja Sander Põldma ja Gerdo Germann selle Electroniga²⁰, mis koondab kõik rakenduse osad ehk eessüsteemi, tagasüsteemi, andmebaasi ja serveriloogika ühte töölaarakendusse.

3.2 Rakenduse teises etapis arendatavad komponendid

Teises arendusetapis keskendus töö autor digiligipääsetavust toetavate ning kasutajakogemust parandavate funktsionaalsuste arendamisele. Rakenduse arendustööd jagunesid mitmeks ülesanderühmaks nagu välja toodud peatükis 1.4.2, millest osa olid väiksema mahuga visuaalsed täiendused, teised aga suuremat analüüsi ja keerukamat arendust nõudvad funktsionaalsused, mis hõlmasid mitut digiligipääsetavuse suunise täitmisnõuet. Alljärgnevalt tuuakse välja olulisemad ja kesksamad arendatud komponendid koos nende eesmärgi, ligipääsetavuse nõuete ja teostuse kirjeldusega.

Lisaks sellele muutis autor ümber ka varasemaid tekstide sõnastusi. Selleks sai töö autor vastava tekstidega sisendi tooteomanikult, mille väljavahetamisel rakenduses tuli tähelepanelik olla, et õiged tekstid läheksid õigesse kohta. Muutmiseks oli vaja asendada väärtuseid nii

¹² W3Schools. HTML Introduction: https://www.w3schools.com/html/html_intro.asp

¹³ W3Schools. CSS Introduction: https://www.w3schools.com/css/css_intro.asp

¹⁴ W3Schools. TypeScript Introduction: https://www.w3schools.com/typescript/typescript_intro.php

¹⁵ Spring. Spring Boot: <https://spring.io/projects/spring-boot#overview>

¹⁶ W3Schools. Java Introduction: https://www.w3schools.com/java/java_intro.asp

¹⁷ SQLite. About SQLite: <https://www.sqlite.org/about.html>

¹⁸ Apache Maven Project. Introduction: <https://maven.apache.org/what-is-maven.html>

¹⁹ Launch4j. Cross-platform Java executable wrapper: <https://launch4j.sourceforge.net/>

²⁰ Electron. What is Electron?: <https://www.electronjs.org/docs/latest/>

eestikeeles kui ka ingliskeelses JSON²¹ failis, mis võimaldab rakenduse keelt vahetada ning loodi ka uusi SQL-päringuid, et muuta andmebaasidesse talletatud kombinatsioonide infot, mida on kirjeldatud peatükis 1.1.

3.2.1 Nupud

Rakendusse oli teises etapis vaja luua uusi nuppe erinevate funktsionaalsustega ning lisaks vaadata üle nupud, mille funktsionaalsus oli loodud teise tiimiliikme poolt. Kõik nupud on loodud semantiliste HTML-nupuelementidena ning nende nuppude märgendite peamise omadusena on igale neist määratud klikkimise funktsionaalsused. Samad funktsionaalsused on edasi antud ka vastavatele klaviatuuriklahvidele, et tagada nuppude ligipääsetavus ka klaviatuuri kasutajatele.

Peitmise ja taas avamise nuppude „Peida” ning „Peida veerg” jaoks sai töö autor implementeerimiseks sisendi tooteomanikult. „Peida veerg” nupp on mõeldud „Kirjeldus, mil määral on eeltingimus täidetud” topelt tulba peitmiseks, sest see topelt tulp sisaldab identset infot, kuid alternatiivsel kujul, mistõttu peab olema kasutajal see võimalik soovi korral peita. Kui veerg on nähtaval, siis asub nupp veeru päises ning tulba peitmisel kuvatakse nupp veeru „Kirjeldus, mil määral on eeltingimus täidetud” päises. Nupu asetuse asukoha soovitus tuli kaasa esimese etapi disainerilt ja tooteanalüütikult. Soovitus analüüsiti teise etapi töö alguses ja leiti, et soovitatud lahendus sobib. Nupu pealdises on lisaks selle funktsionaalsusele esitatud ka kasutatav kiirklahvi kombinatsioon „Peida veerg (Ctrl + H)”, et kasutaja saaks sellest võimalusest kiiresti teada. Kuna rakenduses ei kasutatud liiga palju erinevaid klahvikombinatsioone, puudus vajadus eraldatud infonupu järele ning antud lahendus tagas kõige lihtsama kasutatavuse ja ligipääsetavuse. Joonisel 2 on toodud „Peida veerg” nupp.

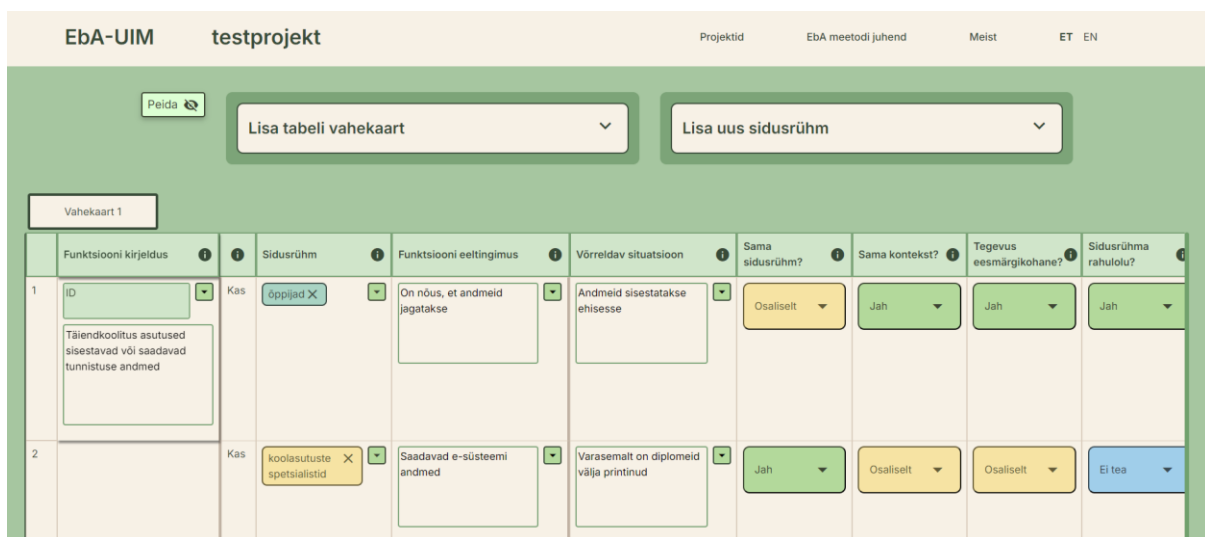


²¹ W3Schools. JSON - Introduction: https://www.w3schools.com/js/js_json_intro.asp

Joonis 2. Veeru „Kirjeldus, mil määral on eeltingimus täidetud” topelt veeru peitmise nupp.

Ekraanitõmmis rakendusest

„Peida” nupp oli mõeldud tabeli kohal asetsevate paneelide „Lisa uus vahekaart” ja „Lisa uus sidusrühm” peitmiseks ja taas avamiseks. Selle nõude eesmärk on vähendada tabeli kohal olevat ruumi, et võimaldada kasutajale suuremat nähtavat tabeliosa lehe üleval. Nupu loomise jaoks saadud sisend eeldas, et täisekraanivaates paiknev nupp on kahest paneelist vasakul, kuid kõik kolm komponenti paiknevad horisontaalselt paralleelselt ning väiksematel ekraanidel koonduvad vertikaalselt nii, et nupp on kõige ülemine element. Joonis 3 kujutab „Peida” nuppu ja paneelide joondust.



Joonis 3. Tabeli kohal asetsevad paneelid ja nupp nende peitmiseks. Ekraanitõmmis rakendusest

Analüüsi tabel võib paljude lisanduvate ridade tõttu muutuda pikaks, mistõttu oli kasulik luua „Üles” nupp. Teiste lehtede eeskujul, nagu Eesti Rahvusringhäälingu uudisteportaali leht www.err.ee ning Tarbijakaitse ja Tehnilise Järelevalve Ameti leht www.ttja.ee leht, paigutati see tabeli alumisse paremasse nurka, et juba olemasoleva kasutaja kogemuse põhjal ei peaks seda uuest kohast otsima. Nimetuses kasutati nii noole märki kui ka teksti, et tagada funktsionaalsuse arusaadavus. Funktsionaalsuse eesmärk on võimaldada kasutajal kiiresti naasta lehe ülaossa.

Answering machine text		Online booking		Reminder					
	Funktsiooni kirjeldus		Sidusrühm	Funktsiooni eeltingimus	Võrreldav situatsioon	Sama sidusrühm?	Sama kontekst?	Teg	ees
8	3 After clicking on the BOOK ONLINE button, the web page displays an encouraging text, which confirms that the customer will be guided when booking	Do	elderly customers	get encouragement from the displayed text to continue.	Encouraging co-texts for the same purpose on other websites elderly customers use.	Osaliselt	Jah		E
9					Encouraging guidance in learning new things or technology that leads to action.	Jah	Osaliselt		C

Joonis 4. „Üles” nupu asend tabeli all paremas nurgas. Ekraanitõmmis rakendusest.

Joonisel 4 on näidatud nupu „Üles” asukoht tabeli vaates. Seoses probleemiga, kus tabel muutub aeglaseks, kui lisatud on liiga palju ridu, mida on kajastatud kompleksarendaja bakalaureusetöös, otsustati tabeli kuvamisviisis teha muudatused [29]. Muudatused hõlmasid tabelivaate liigutamist spetsiaalsesse Angular konteinerisse *cdk-virtual-scroll-viewport*²², mis kuvab ainult nähtavaid ridu pikast loetelust, et optimeerida tabeli jõudlust. Sellisel kujul lahendus lisas tabeli jaoks eraldi vertikaalse keritava ala lisaks paralleelselt kogu lehe kerimisvõimalusega. Kuna tabelis on palju veerge, siis on see ka horisontaalselt keritav ning seetõttu ei olnud võimalik „Üles” nuppu paigutada tabeli konteinerisse vaid väljapoole seda, et tagada nupu pidev kuvamine, kui selleks on tekkinud vastav olukord.

Uutele nuppudele lisaks oli tarvis ka üle vaadata kompleksarendaja poolt loodud nupud, millega sai üles laadida JSON formaadis faili ning laadida projekt alla Excel või JSON formaadis. Mõlemal juhul olid nupud juba ühtlase kujunduse ja sobiva paigutusega. „Lae üles” nupule lisas töö autor vaid sellele iseloomuliku ikooni ning lisas nimetusse juurde faili formaadi ehk lõplikuks nimetuseks kujunes „Lae üles (JSON)”.

3.2.2 Näidete prioritseerimine

EbA analüüsi meetodi täitmise viies etapp pakub kasutajale võimalust prioritseerida näiteid, mida on kirjeldatud peatükis 1.1. Kui tekib olukord prioritseerimiseks ehk ühele eeltingimusele lisatakse mitu näidet, avaneb pärast tulba „Sidusrühma rahulolu?” kõrvale uus tulp, mis on mõeldud prioritseerimise funktsionaalsuseks. See on lahendatud märkeruudu abil,

²² Angular Material. cdkScrollable and ScrollDispatcher: <https://material.angular.dev/cdk/scrolling/overview>

kus prioritseerimist vajavate näidete ridadele vastavasse tulpa tekib märkeruut, mille ära märkimisel on vastaval real olev funktsiooni eeltingimuse näide prioriteetne. Read ehk näited, mis ei ole märkeruuduga märgitud, muutuvad mitte prioriteetseteks ning seda on edasi antud visuaalse lahendusega, kus vastava näite rea taust alates prioriteedi tulbast kuni „Tee järeldused ja loo tegevuskava” tulbani kuvatakse halli taustatooniga.

Funktsiooni kirjeldus	Rühma olemine?	Kirjeldus, mil määral on eeltingimus täidetud	Peida veerg (Ctrl+H)	Tee järeldused ja loo tegevuskava
1 Täiendkoolitus asutused asustavad või saadavad tunnistuse andmed	<input type="checkbox"/>	Osad Õppijad On nõus, et andmeid jagatakse	Osad õppijad On nõus, et andmeid jagatakse	Kategoriseeri tegevusplaan
2 Osaliselt <input type="checkbox"/> Ei tea <input type="checkbox"/>	<input checked="" type="checkbox"/>	Veidi teises kontekstis pole rahulolu selle eeltingimusega teada, kuid tolles veidi teistsuguses kontekstis mõningal määral	Veidi teises kontekstis pole rahulolu selle eeltingimusega teada, kuid tolles veidi teistsuguses kontekstis mõningal määral koolasutuste spetsialistid Saadavad e-süsteemi andmed	Kategoriseeri tegevusplaan
3 Osaliselt <input type="checkbox"/> Ei tea <input type="checkbox"/>	<input type="checkbox"/>	Rahulolu eeltingimusega pole teada, kuid mõningal määral osad	Rahulolu eeltingimusega pole teada, kuid mõningal määral osad koolasutuste spetsialistid Saadavad e-süsteemi andmed	Kategoriseeri tegevusplaan

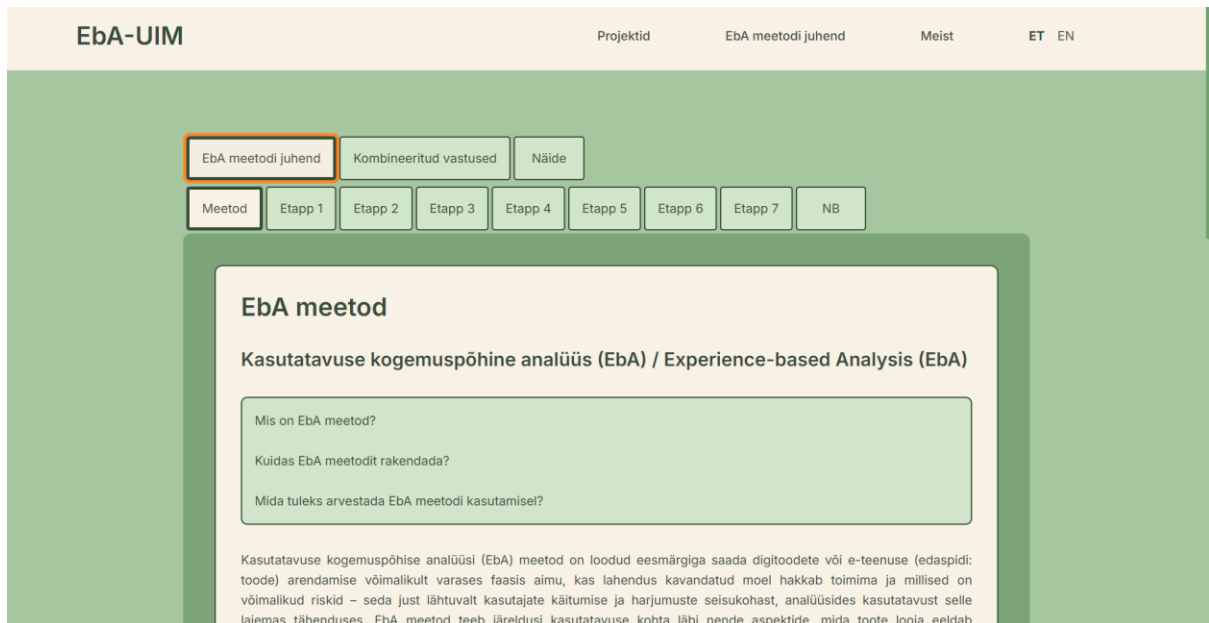
Joonis 5. Prioriteetse rea määramise märkeruudud ning tulemuse visuaal. Ekraanitõmmis rakendusest

Oluline oli mitte prioriteetse näitega rea puhul tagada, et tekst oleks loetav ning et kaetud oleks ka tulpa „Tee järeldused ja loo tegevuskava” kategoriseerimise värv. Prioritseerimise võimaluse lahendus ja selle tulemus on kuvatud joonisel 5.

3.2.3 Klaviatuuriga navigeerimine

Rakenduse kasutamise võimaldamiseks klaviatuuriga oli vaja sisse viia mitmeid muudatusi eessüsteemi koodis. Algselt oli klaviatuuriga navigeeritav vaid menüüriba ning osaliselt ka valitud projektisisene tabel. Tabelis oli võimalik liikuda mööda interaktiivseid sisestusvälju. Fookusena oli vaikimisi peenike oranž riba ümber interaktiivse elemendi.

Töö autor alustas klaviatuuri ligipääsetavuse parandamisega lehel „EbA meetodi juhend”, kus mööda kahetasandilist vahelehtede süsteemi on võimalik navigeerida erinevat tüüpi info vahel. Esimene ehk ülemine tasand määrab teema, milleks on „EbA meetod”, „Kombineeritud vastused” või „Näide” ning vastavalt teema vahelehe valimisest kuvatakse alumisel vahelehe tasandil selle teema alajaotused ja neist vaikimisi avaneb esimene. Joonisel 6 on näha vahelehtede süsteemi, kus fookus on “EbA meetodi juhendil”.



Jooni 6. Lehe „EbA meetodi juhend” vahelehtede süsteem ning klaviatuuri fookus.

Ekraanitõmmis rakendusest

Vahekaartide süsteemi jaoks oli selle lehekülje HTML-is kasutatud Angular raamistiku komponenti *mat-tab-group*, mis võimaldab luua vahekaartidega kasutajaliidest ning mis juba järgivad WAI-ARIA vahekaartide mustrit²³ [31]. Rakenduse esimeses versioonis polnud klaviatuuriga navigeerimise tagamine eraldi eesmärgiks, mistõttu ei vastanud vahekaartide vahel liikumine veel vastavale mustrile. See ei toimunud põhjusel, et vahekaardi komponendile vastav CSS reegel oli sellele omase funktsionaalsuse kaotanud ülekirjutamise tõttu. Selle asendamisel õige CSS reegluga, saavutas vahekaartide süsteem WAI-ARIA mustrile omase navigeerimise.

Vahelehe „Projektid” puhul ei toimunud sarnaselt teistele lehtedele klaviatuuriga navigeerimine. See lisati projekti lisamise paneelile, projekti sisenemise nupule ning projekti lisavõimaluste menüüle, andes sisestus- või tühikuklahvile samasuguse funktsionaalsuse, mis hiireklikil on.

Projektisisises vaates, mis kujutab tabelit, oli suurem osa lehest klaviatuuriga ligipääsetav. Kõik komponendid, mis olid HTML koodis semantiliselt märgendatud nagu näiteks nupud ja sisestusväljad, olid edukalt klaviatuuriga navigeeritavad. Tabeli kohal olevatele paneelidele pidi sarnaselt „Projektid” lehele andma vastavad funktsionaalsused sisestus- või tühikuklahvile. Paneelide all ja tabelist üleval on ka vahekaardi süsteem tabeli erinevate

²³ Angular Material. Tabs: <https://material.angular.dev/components/tabs/overview#a11y>

vahelehtede loomiseks ning nendel pidi korrigeerima CSS reegleid, nagu pidi tegema lehel „EbA meetodi juhend”.

Suurim väljakutse tabeli puhul olid veergudes „Funktsiooni kirjeldus”, „Funktsiooni eeltingimus” ja „Võrreldav situatsioon” olevad rippmenüüd. Need olid varasemalt loodud ilma ühegi teegi või raamistiku lahenduse abita. Kuigi sama menüüd oli kasutatud „Projektid” lehel projekti lisavõimaluste menüüna ning ka tabeli „Sidusrühm” veerus, siis vaid nende kolme eelnevalt välja toodud veeru puhul avanes menüü nii, et fookus jäi sinna lõksu ehk ei liikunud enam edasi. Antud töö autor proovis läbi erinevaid lahendusi selle parandamiseks, kuid ei selgunud, mis seda põhjustas. Teise etapi arenduses leidis autor, et ainuke viis rippmenüüde ligipääsetavuse tagamiseks on neile kolmele veerule eraldiseisvate rippmenüü komponentide loomine veidi lihtsama ja veakindla lahendusega. Kuna loodud lahendus ei ole heade tavade järgi (ingl *best practices*) loodud, on kasulik seda probleemi ja komponentide funktsionaalsust analüüsida järgmises etapis, et tuvastada võimalikke parenduskohti.

Navigeerida polnud võimalik ka tabeli veergudes olevatel infonuppudel, mis on aga kasutaja jaoks väga oluline. Selle lahendamiseks oli vaja olemasolevale Angular komponendile, mis kuvas infotähise ikooni, lisada nupu funktsionaalsus.

Lisaks oli tabelivaates kasutusel „Peida veerg” nupp, millele loodi kahe klahvi kombinatsioon lisaks klikkimisele, et see aktiveerida. Sellest inspireerituna loodi ka „Üles” nupule kahe klahvi kombinatsioon juhtklahvi (ingl *Ctrl key*) ja ülesnooleklahviga. Kuigi esimesel nupul on kombinatsioon nimetusse lisatud, otsustati seda mitte teha „Üles” nupu puhul ning see info antakse edasi rakenduse lehel „Meist”. Isikliku lisana nendele kombinatsioonidele tehti võimalus neid vajutada ükshaaval. Koodis on määratud 3-sekundit, mis hakkab käima alates juhtunupu vajutamisest ning kolme sekundi jooksul ootab H- või ülesnooleklahvi vajutust. Selle lahendusega mõtles autor nende peale, kellel ei ole võimalik kasutada kahte sõrme korraga klahvikombinatsiooni sisestamiseks.

3.2.4 Ekraanilugeri võimaldamine

Käesoleva bakalaureusetöö üks eesmärkidest on edendada ekraanilugeri toe võimalust. Töö autor kasutas koos koodi muudatuste tegemisega tarkvara JAWS²⁴, et kontrollida selle

²⁴ JAWS: <https://www.freedomscientific.com/products/software/jaws/>

töötamist. JAWS on eraldiseisev rakendus ning selle kasutamisel pole vaja peale käivitamise midagi eraldi ühildada, sest rakendus oskab ise tuvastada kasutaja tegevusi arvutis.

Esimeses etapis loodud eessüsteemi kood oli vastavalt HTML semantilistele elementidele märgendatud. Tänu sellele on ekraaniluger võimeline lehel teksti lugema ning seda ka erinevatelt interaktiivsetelt elementidelt. Kuna ekraanilugeri võimaldamisega tegeleti paralleelselt koos klaviatuuri navigeerimise tagamisega, lahenesid sellega koos kohad, kus ekraaniluger eelnevalt infot ei suutnud lugeda.

Sellest hoolimata oli vaja juurde lisada koodi *aria-label*²⁵ atribuudid, mis on justkui ligipääsetavad nimed ning need vajalikud kohtades, kus ei pruugi olla teksti kujul nimetust vaid näiteks ikoonid. EbA-UIM rakenduse koodis on laadimist kujutav sümbol, millele tuli anda ligipääsetav nimi, mis annab märku, et leht veel laeb infot. Nii ei teki kasutajal segadust, et miks ei loe ekraaniluger sisu ette. Samuti on sellise nimede lisamine oluline kui mingeid elemente saab sulgeda risti või mõne muu sarnase ikooni abil. Selliseid lahendusi esines ka EbA-UIM rakenduses, kuid need märgendati ära ligipääsetavate nimedega vastavalt.

3.2.5 Teavitused ja märguanded

Rakenduse peamine funktsionaalsus on mõeldud analüüsi teostamiseks tabelis. See sisaldab aga mõningaid tegevusi, mida on vaja üle kontrollida, et ei tehtaks vigu. Ridadel on erinevaid kustutamise ja ka teisi ülejäänud analüüsi mõjutavaid tegevusi. Sellistes olukordades, nagu näiteks on kustutamine, tuli kasutajale esitada märguanne tegevusest ja küsida kinnitust. Selleks kasutati esimese etapi disaineri loodud samalaadseid märguandeid, et tagada visuaalne ühtivus sarnaste tegevuste puhul.

Lisaks sellele oli vajalik ka implementeerida esimese etapi disaineri kujundatud märguanded veergu „Tee järeldused ja loo tegevuskava”. Märguande esinemise eeltingimuseks oli ühe rea ehk ühe funktsiooni eeltingimuse näite kohta kõikide etappide täitmine, mida on kirjeldatud peatükis 1.1. See tähendab, et ka veeru „Tee järeldused ja loo tegevuskava” lahter oli täidetud. Koodis oli vaja teostada kontroll, et kui eelnevalt mainitud veeru lahtris olev sisestusväli oli täidetud ning pärast seda muudetakse nelikvastuseid, mis on analüüsi neljas etapp, siis tekib veeru „Tee järeldused ja loo tegevuskava” lahtri sisestusvälja kohale märguanne. See annab teada, et eelnevalt on toimunud muudatus, mistõttu võivad eelnevad sisestatud järeldused ja

²⁵ MDN Web Docs. ARIA: aria-label attribute: <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-label>

tegevuskava vajada täiendusi. Tooteomaniku soovist tekkis ka vajadus salvestada märguanne andmebaasi, et seda säilitada erinevate kasutussessioonide vahel, kui kasutaja pole märguannet kinni pannud. Selle jaoks lõi kompleksarendaja andmebaasi märguannet esindava veeru ning mõlemad nii kompleksarendaja kui ka käesoleva töö autor löid ettenähtud muudatused vastavalt tagasüsteemis ja eessüsteemis.

3.2.6 Arendusest välja jäänud nõuded

Teise arendusetapi nõuete seas oli neid, mis tooteomanikuga kokkuleppel jäid vaid analüüsi tasandile ning arendusse minek otsustatakse järgmises arendusetapis. Üheks selliseks funktsionaalsuseks oli ridade omavaheline vahetamine. Töö autori esmane välja pakutud idee oli kasutada ridade vahetamiseks *drag-and-drop* lahendust, mis eeldab, et vahetataval real on ankur, millele hiirega saab vajutada ning peal hoides on võimalik rida lohista uuele sobivale reale. Küll aga pole selline lahendus ligipääsetav kõigile, sest ei saa eeldada, et kõik kasutajad on võimelised vajutama, hoidma peal ja liigutama hiirt samaaegselt. Sellest lähtuvalt, kui luua taoline lahendus, eeldab see ka alternatiivse viisi loomist, sest ka klaviatuurikasutajatele pole lohistamisega seotud lahendused ligipääsetavad.

Vahekaart 1											
	Funktsionaalsuse kirjeldus	Sidurühm	Funktsionaalsuse eeltingimus	Võrreldav situatsioon	Sama sidurühm?	Sama kontekst?	Eesmärgipärane kasutus?	Lahendusega rahul?			Kirjeldus, mil määral on eeltingimus
1	ID	Kas	Testija X	2	3	Jah	Osaliselt	Ei	Ei tea		Teisiti ei ole võimalik teada, kas kontekstis on ebakorrektne, lisaks pole teada, milline on rahulolu eeltingimusega ja kas veidi teistsuguses kontekstis Meil pole alust eeldada, et
2				4	Jah	Osaliselt	Osaliselt	Ei			Väga väheses kontekstis ei ole selle eeltingimusega rahul, kuid selles veidi teistsuguses kontekstis mõningal määral
3		Kas	Testija X	10	11	Ei tea	Ei tea	Jah	Jah		On mingi sidurühm, kes mingisuguses teadmata kontekstis tegutseb eesmärgipäraselt, kuid me ei kas
4	ID	Kas	Proovija X	6	9	Osaliselt	Ei tea	Jah	Jah		Mingisuguses teadmata kasutuskontekstis osad

Joonis 7. Vahetust vajavad read. Ekraanitõmmis Figmast

Joonisel 7 on visuaalse ülevaate andmiseks ka näidatud, missugusel viisil on vaja võimaldada ridade vahetust. Tumesinisega on tähistatud funktsiooni väli. Funktsioon hõlmab eeltingimusi, mis on tähistatud oranžiga. Võrreldav situatsioon on roosaga tähistatud ning kuulub omakorda eeltingimuste alla. Pöörates tähelepanu disainilisele poolele, eeldab sellisel kujul reavahetuse

teostamine iga välja toodud grupi puhul, ükskõik millise lahenduse korral, selgelt eristatavaid indikaatoreid või ankruid. Kui aga ridu on palju, sealhulgas eeltingimusi ja näiteid, koormab see visuaalselt tabeli üle. Seega on selle funktsionaalsuse jaoks vaja teha kitsendusi. Üks võimalus on teostada lahendus, kus vahetada saab omavahel ainult funktsioone, mida joonisel on kujutatud tumesinise plokina. Lisaks vähendaks ühe tugeva lahenduse loomine arenduse koormust ja müra kasutajaliideses. Üheks võimalikuks lahenduseks oleks funktsiooni ridadele implementeerida noolenupud, mida vajutades saab liigutada vastavat funktsiooni ühe funktsiooni võrra üles või alla.

Teine analüüsitud nõue käis ridade vahele lisamise kohta. See tähendab, et on võimalik lisada uus funktsioon valitud reale tabelis. Keerukuselt langeb see sarnasele tasemele nagu ridade vahetamise funktsionaalsus, sest eeldab erinevate seotud plokkide kontrollimist, et ei tekiks tegevusi, mis rikub tabeli loogikat. Selle nõude lahenduseks on võimalik võtta inspiratsiooni juba olemasolevast tabeli funktsionaalsusest. Veergude „Sidusrühm”, „Funktsiooni eeltingimus” ja „Võrreldav situatsioon” lahtrites on rippmenüüd, mille valikus on vastavalt igale veerule võimalus lisada uus selle veeru komponent. Kui kasutajatel on juba olemas sellise funktsionaalsusega lahendus ja selle kasutamise kogemus, oleks järjepidevust arvesse võttes kasulik teha ka uue rea ehk funktsiooni lisamise võimalus veeru „Funktsiooni kirjeldus” lahtri rippmenüüsse.

Kolmandaks toodi välja vajadus märkmete lisamiseks nelikvastuste tulpa, et analüüsi täpsustada. Kuna rakenduse tabel hõlmab oma funktsionaalsuse täitmiseks paljusid komponente, tuleb uute lisamine väga läbi mõelda. Algse lahendusena pakuti välja vastavatesse lahtritesse paigutada märkmeid sümboliseeriv ikoon, millele vajutades avaneks sisestusväli, kuhu saab kirjutada, mis salvestab teksti ning saab seejärel sulgeda. Sellisel juhul peaks pärast märkme tegemist muutuma ka ikoon või lisanduma uus märgeline lahtrisse, et näidata loodud märkme olemasolu. Detailsema ikooni lisamisel tekib jällegi tabelisse palju müra, eriti kui neid on iga rea vastavas veerus. Sellisel juhul on raske tuvastada, kuhu märgeline on tehtud ja kuhu mitte. Koos tooteomanikuga leiti võimalik lahendus, mis on samal põhimõttel nagu töötab Microsoft Exceli tabelis märkme lisamine. Kasutaja saab vajutada parema hiireklikiga või klaviatuuril ettenähtud klahvikombinatsiooni, mille järel avaneb võimalus lisada lahtrisse märgeline. Kuna rakendus salvestab pidevalt sisestatud infot, pole vajadust eraldi salvestusfunktsioon loomiseks. Sellisel juhul on märkmete tegemise kohta väga vajalikud vastavad juhised infonupukestes, et kasutaja oleks selle funktsionaalsusest teadlik. Sellisel kujul lahendus pole ideaalne, sest puudub visuaalne indikaator, kuid lähtutud on selle puhul

levinud põhimõtetest ja võimalikul varasemast kogemusest. Nõuet saab edasianalüüsida vajadusel parema lahenduse leidmiseks ning implementeerimiseks järgmises etapis.

Lisaks analüüsiti ka olemasoleva „Lisa uus funktsioon” nupu asetust. Nupu praegune asukoht, mis määrati esimese etapi arenduses, on loogiliselt tabeli all, kuhu tekivad lisamisel uued read. Kaaluti nupu viimist tabeli ülemisse vasakusse serva esimese etapi disaineri disaini põhjal, kuid see ei olnud intuiitiivne lahendus. Uue rea lisamise funktsionaalsust kataks ka rea vahele lisamise võimaluse tagamine, mida eelnevalt on kirjutatud. See tagaks mitu viisi uue rea lisamiseks. Seetõttu säilitati nupu asukoht tabeli all vasakul pool ning järgmises arendusetapis on võimalik teostada rea vahele lisamine, mis loob kasutajale rohkem võimalusi efektiivsemaks tööks.

4. Tulemused

Selles peatükis kajastatakse töö autori testimise tulemusi, kuid osaliselt ka tarkvara testija tehtud testimisi. Lisaks antakse ülevaade valminud rakenduse andmetest.

4.1 Testimine

Rakenduse EbA-UIM teise etapi arenduses viidi läbi testimised, mille eesmärk oli hinnata nii loodud lahenduste funktsionaalsust kui ka nende vastavust ka peatükis 1.4.3 välja toodud digiligipääsetavuse suunistele. Teostati kolme tüüpi testimisi: üldine kasutatavuse testimine, klaviatuuri kasutatavuse testimine ning manuaalne WCAG kriteeriumite põhine kontroll. Üldise kasutatavuse testimise tegi tarkvara testija ning tema viis läbi ka klaviatuuri kasutatavuse testimise. Töö autori poolt tehtud WCAG nõuetele vastavuse testimine võimaldas tuvastada probleeme ning kontrollida, kas loodud lahendused vastavad ligipääsetavuse või kasutusmugavuse nõuetele. Alljärgnevates alapeatükkides on lähemalt selgitatud, kuidas testimised toimusid, mida selleks kasutati ning millised olid tulemused.

4.1.1 Üldine kasutatavuse testimine

Tarkvara testija viis läbi üldise kasutatavuse testimise, mida on põhjalikumalt kajastatud ka tarkvara testija bakalaureusetöös [30]. Selleks koostas tarkvara testija stsenaariumid, mille põhjal saaks testides rakenduse funktsionaalsustest ja nende toimimisest hea ülevaate. Testimisel osales viis testijat, kellest mõni oli varasemalt kokku puutunud nõuete analüüsimisega, kuid oli ka neid, kes sellega tuttav ei olnud. Järgnevates lõikudes tuuakse välja testimiselt selgunud tulemused, mis on seotud käesoleva töö autori loodud lahendustega või puudutavad osi, mille parandamine on tema ülesandeks määratud.

Testimise tulemustest selgusid mõned kohad, mis vajavad parandusi. Näiteks tekkis rakenduses olukord, kus kadus ära alumine horisontaalne kerimisriba. Kuna tabelivaade oli vaateväljas ära lõigatud täpselt kolmikvastuse tulba servast ja ei olnud näha järgmise tulba algust, jäi paarile testijale mulje, et sellega tabeli vaade lõppeb. Kuna kerimisriba polnud näha, siis ei olnud ka mingit indikaatorit, mis hoolimata kerimisriba puudumisest ajendaks kasutajat siiski horisontaalset kerimist proovima. Antud probleemi esmaseks lahenduseks tuleb teise arendusetapi jooksul teha tabeli vaateväli natuke laiemaks, et kasutaja näeks edasise tulba algusosa, mis viitaks, et paremal pool tabel jätkub.

Teiseks selgus, et „Tee järeldused ja loo tegevuskava” tulba kategoriseerimiseks mõeldud värvid olid vales järjekorras. Kui hea tava järgi algaks kategoriseerimine alates positiivsest kuni negatiivse või neutraalse kategooriani, siis testimisel kasutatud versioonis oli esimesena neutraalne kategooria, millele järgnes õige järjestus. Järjestust parandati testimise järgselt.

Ebaühtlane käitumine tekkis „Üles” nupu puhul, mis mingitel hetkedel, olles tabeli lõpus, ei ilmunud ekraanile. Lehel on kaks keritavat ala ning tabeli jaoks on kasutatud *cdk-virtual-scroll-viewport-i*, mis kuvab ainult need elemendid, mis paiknevad nähtavas vaateväljas või selle läheduses. Seega paigutati testimisel kasutatud versioonis „Üles” nupp välimisse vaatevälja, et see jäädavalt ära ei kaoks ekraanilt. Testimise tulemus näitas, et lahendus oli vaja teise etapi arenduse jooksul üle vaadata ning parandada.

Tabelis on veerg „Kirjeldus, mil määral on eeltingimus täidetud”, mille sisu on moodustatud kolmest osast ning need on jagatud tulba sees kolme veergu, et see moodustaks terviklikult loetava lause. Testimisest selgus, et leiduvad olukorrad, kus lauseosad jooksevad üle veeru ning seega võib katta ka järgmise tulba infot. Lisaks selgus ka vajadus tulba erinevate osade joonduse muutmiseks, et parandada loetavust. Veeruga seotud probleemid ja täiendused on kavas parandada teise arendusetapi jooksul.

Viimaseks toodi välja näidete prioritseerimise tulemused. Kui ühele näitele lisatakse mitu näidet, millele tekivad eri vastuste kombinatsioonid neljandas etapis, mida kirjeldab peatükk 1.1, tekib võimalus neid näiteid prioritseerida. Selle olukorra tekkest annab teada ka vastav teavituse kuvamisel ja prioritseerimise võimaluse tekkimisel on aga väga oluline jälgida just erinevate kombinatsioonide tekkimist. Testimise käigus selgus, et teavituse ilmumine kohe pärast teise näite lisamist ühele eeltingimusele, mis ei ole õige käitumine. Teavituse kuvamisel on vaja teha parandusi, mis vastaks eelpool toodud kirjeldusele ning need teostatakse teise etapi arenduse jooksul.

Kokkuvõtlikult selgus, et rakendus oli kasutatav, kuid esines ebamugavusi, mida saab parandada. Kõik eelnevates lõikudes välja toodud kitsakohad anlüüsitakse testimise järgselt ning parandatakse teise arendusetapi jooksul.

4.1.2 Klaviatuuriga navigeerimise testimine

Klaviatuuriga kasutatavuse testimise eesmärk selle töö raames oli kontrollida klaviatuuriga navigeerimist EbA-UIM rakenduses. Testimise jaoks oli vajalik töö autori sisend tarkvara testijale, kes testimist korraldas ja läbi viis. Sisendiks koostas töö autor kasutajale mõeldud

ülesanded, mis põhinesid realistlikul funktsioonide läbimisel rakenduses klaviatuuri abil. Lisas 3 on esitatud testimisel kasutatud ülesanded koos ülesannete oodatud lahendamise ja tulemuste kaardistamiseks mõeldud tabeli malliga.

Töö autor osales testimisel jälgijana, mis võimaldas dokumenteerida erinevaid probleeme ja tähelepanekuid, mis klaviatuuriga navigeerimist piiravad või raskendavad. Lisaks ilmsid mitmed positiivsed aspektid. Testijal, kes oli kutsutud klaviatuuri kasutatavuse testimisel osalemiseks, ei tekkinud olukorda, kus klaviatuuriga navigeerides oleks fookus mingile elemendile kinni jäänud. Samuti oldi rahul klaviatuuri fookuse visuaaliga ehk oranži raamiga ümber elementide, millel fookus parasjagu oli. Kokkuvõtlikult oli kogu rakendus edukalt navigeeritav, kuid tuvastati ka järgmised kitsaskohad, mis muutis klaviatuuriga navigeerijate ligipääsetavust märgatavalt ebamugavaks:

1. Informatiivsel ja lugemiseks mõeldud “Meist” lehel tekkis ebavajalik linkide läbimine enne soovitud nupuni jõudmist. Lehel tekib nähtavale “Üles” nupp, kui kasutaja on jõudnud lehe alumisse osasse, kuid selle kasutamiseks pidi kasutaja läbima tabulaator nupuga kaks teksti sees olevat meiliaadressi linki, enne kui sai fookuse soovitud nupule suunatud.
2. Fookus kadus osade funktsionaalsuste kasutamise järgselt. See tähendab, et näiteks tabeli lõpus oleva nupu abil uue rea lisamise korral tekib edukalt uus rida, kuid fookus kaob ning ilmub lehe algusesse uuesti alles pärast tabulaator klahvi vajutust. Sama probleem esines ka veeru peitmiseks mõeldud nupu puhul või tabelis olevate lisamenüüde kasutamisel veergudes “Funktsiooni kirjeldus”, “Funktsiooni eeltingimus” ja “Võrreldav situatsioon”.
3. Mõne tabeli funktsionaalsuse kasutamisel liikus funktsionaalsus ebamugavasse kohta. Näiteks rea lisamisel, selle asemel, et fookus liiguks uuele reale, liikus see hoopis lehe algusesse. Uue eeltingimuse lisamiseks saab kasutada olemasoleva eeltingimuse veerus olevat menüüd, kus on selleks vastav tegevus, mille kasutamisel tekib tabelisse uus eeltingimus, kuid fookus jätkab selle tegevuse järel algse eeltingimuse menüüst järgmisel elemendil. Samuti liigub fookus lehe algusesse ridade kustutamisel, selle asemel, et suunata fookus kustutatud reast üleval või allpool olevale reale.
4. Veerus “Sidusrühm” pärast sellele omase menüü avamist, pidi selle manuaalselt ise sulgema ehk menüü jäi lahti pärast menüült klaviatuuriga väljumist ega polnud võimalik sulgeda seda paoklahvi (ingl *Escape*) abil.

Tagasiside analüüsitakse koos vastavate muustrite põhjal läbi ning otsustatakse, mis kujul lahendused tehakse. Nimetatud probleemid on kavas lahendada teise arendusetapi jooksul.

4.1.3 Ligipääsetavuse manuaalne testimine

Ligipääsetavuse põhjalik hindamine toimus manuaalselt vastavalt lisas 1 välja toodud testimisse võetud kriteeriumitele. Selleks koostas töö autor ülevaate lahendustest ja millised kriteeriumid neile vastavad ning see on välja toodud lisas 4. Tabelis on välja toodud iga arenduses teostatud lahendus ja märgitud ära sellele vastavad digiligipääsetavusnõuded. Manuaalne testimine viidi läbi WCAG kriteeriumite puhul vastavalt WCAG 2.2 juhistele ning see hõlmas nii koodi visuaalset ja semantilist kontrolli kui ka kasutajaliidese testimist hiire ja klaviatuuriga. Kriteeriumite täitmist hinnati lähtudes nende ametlikust täitmisnõuetest. Testimise käigus kontrolliti interaktiivsete elementide ligipääsetavust klaviatuuriga ning veenduti, et kontekst või fookus ei muutuks ootamatult ilma kasutaja tegevuseta. Ülevaatuses alla kuulus ka sisuelementide korrektne semantiline tähistus ning ARIA-atribuutide rakendamine. Nende puhul vaadeldi otseselt HTML- ja Angular-komponentide lähtekoodi. Lisaks kontrolliti ka värvikontraste vastavalt WCAG nõuetele ning selleks kasutati Accessible Web värvikontrastsuse kontrollimise tööriista²⁶.

Testimise käigus selgus kaks kriteeriumi, mis polnud täielikult täidetud. Esimene neist on WCAG 2.2 kriteerium 1.4.13 ehk „Sisu lohistamisel või fookuses”²⁷, mille eesmärk on tagada, et ajutiselt kuvatav sisu ei kaoks ootamatult, näiteks hiire fookuse eemaldamisel elemendilt. Rakenduse esimeses arendusetapis loodi veergude päisesse infonupud, mis avanevad, kui infonupul hiirega üle lohistada ja sulguvad, kui kursor sellelt ära liigutatakse. Teises arendusetapis võeti fookusesse ka klaviatuuriga navigeerimise võimaldamine ning antud infonupud esimese etapi järgselt ei olnud klaviatuuriga ligipääsetavad. Seetõttu kohandati olemasolevat lahendust, et seda võimaldada, kuid sellega ei tagatud hiire puhul infonupu sisu püsivat lahti jäämist. Teises arendusetapis ei leitud sellele paremat lahendust, et tagada samaaegselt nii klaviatuuri fookuse võimalus kui ka hiirega püsivalt infonupu avamise ja sulgemise võimalus. Lahenduse leidmine jääb potentsiaalsesse järgmisesse etappi, kus tuleks analüüsida Angular komponentide erinevaid võimalusi või täiesti uusi lahendusi.

²⁶ Accessible Web. WCAG Color Contrast Checker: <https://accessibleweb.com/color-contrast-checker/>

²⁷ W3C. WCAG 2.2 Content on Hover or Focus (Level AA): <https://www.w3.org/WAI/WCAG22/Understanding/content-on-hover-or-focus.html>

Teine tuvastatud probleem ei vastanud kriteeriumile 2.4.13 ehk „Fookuse välimus”²⁸, mille kohaselt peab klaviatuuriga navigeerimisel olema fookuse laius 2 pikslit ning fookuseta oleku ja fookuses oleku vahel peab olema kontrastsussuhe vähemalt 3:1. Näiteks kui interaktiivsel nupul pole fookust, on see hall, aga fookuse saavutamisel tekib sellele kollane ääris, siis fookuse ääris peab varasemast olekust eristuma kontrastussuhtega 3:1. Selle töö raames lahendati fookus kahepikslise äärisega, mis oli neon oranži värvi, kuid mõne elemendi puhul ei saavutanud see nõutud kontrastsust. Olukorra parandamiseks muudeti värv erksa lilla vastu. See saavutas nõutud kontrastsussuhte enamiku eelnevalt probleemsete elementide jaoks, välja arvatud infonuppude korral. Katsetati ka teisi variante kriteeriumi vastavuse saavutamiseks, kuid kuna infonupp on tumeroheline ja heledal taustal, oli raske leida tooni, mille kontrastsussuhe ei läheks teiste erinevate elementidega uuesti vastuollu. Klaviatuuriga navigeerimisel liigub fookus infonupule, mis automaatselt avab ka selle selgitava sisu, siis indikeerib see mõnevõrra siiski kasutaja asukohta. Lisaks on fookuse äärise ja infonupu vahel väike vahe, mis võimaldab hoolimata värvist seda siiski mingil määral tuvastada. Rakenduse järgmises etapis tasub analüüsida uusi võimalike värvilahendusi nii fookuse kui infonupu jaoks.

Eraldi tähelepanu pöörati ka nelja välja toodud EN 301 549 kriteeriumitele. Kontrolliti, kas arendatud kasutajaliides võimaldab funktsioonide aktiveerimist, säilitab rakenduses ligipääsetavuse toe ning kas pakutav dokumentatsioon vastab ligipääsetavuse info edastamisele. Kuna rakendus võimaldab klaviatuuriga navigeerimist ja pakub ekraanilugeri tuge ning neid saab kasutada endale sobival viisil ja hetkel. Rakendus ei lähe vastuollu ega katkesta digiligipääsetavuseks vajalike tehnoloogiate kasutamist. Need omadused täidavad kriteeriumid EN 5.2, EN 11.6.1 ja EN 11.6.2. Kuna rakendus pakub digiligipääsetavuse tagamiseks abitehnoloogiate ühilduvust ning teise arendusetapi jooksul dokumenteeritakse juhised nende kasutamiseks, siis saavutame selle täitmisel ka kriteeriumi 12.1.1.

4.2 Rakenduse valminud tulemus

EbA-UIM rakenduse lähtekood on avalik ning seda hoiustatakse Bitbucket repositooriumis²⁹. Lähtekood on tervik meeskonna lahendustest, kuid igapäevane panust on võimalik näha

²⁸ W3C. WCAG 2.2 Focus Appearance (Level AAA): <https://www.w3.org/WAI/WCAG22/Understanding/focus-appearance.html>

²⁹ Bitbucket. EBA-method: <https://bitbucket.org/eba-method/eba-code/src/master/>

versioonihaldussüsteemi Git ajaloo kaudu. Autori disainilahendused on esindatud Figma³⁰. Rakendust on võimalik alla laadida³¹. Rakendus on kasutatav nii Windowsis, MacOSil kui ka Linuxil.

³⁰ Figma. EBA-UIM: <https://www.figma.com/design/OSxPiJeNO9RM2BPwWQ0meT/EBA-UIM?m=auto&t=0yMvG431Us9VVSe1-1>

³¹ Bitbucket. EbA-method Downloads: <https://bitbucket.org/eba-method/eba-code/downloads/>

Kokkuvõte

Bakalaureusetöö eesmärk oli edasi arendada EbA-UIM rakenduse kasutajaliidest ja eessüsteemi, keskendudes digiligipääsetavust parandavatele lahendustele ning uute funktsionaalsuste loomisele. Rakendus töötati välja meeskonnatöona, kus töö autor panustas uute komponentide visuaalse disaini kujundamisse, eessüsteemi arendusse ning WCAG versiooni 2.2 ning EN 301 549 kriteeriumitest lähtuvate nõuete rakendamisse.

Töös anti ülevaade EbA meetodist, digiligipääsetavuse olulisest ning rakenduse arenduse etappidest. Disainilahendused loodi Figma keskkonnas, arvestades ligipääsetavuse ja kasutusmugavuse põhimõtteid. Töö keskmes olevad lahendused pidid toetama klaviatuuriga navigeerimist ning pakkuma vajadusel ekraanilugeri tuge.

Rakenduse arendamisel kasutati Angulari raamistikku, mis eeldas nii HTML, CSS kui ka TypeScripti kasutamist. Arendust toetati tiimitöö kaudu, milles iga liige täitis selgelt määratud rolli. Eessüsteemi arenduse ülevaate annab alapeatükk 3.2 ning selle toimimist toetas nõuete analüüs ja testimised.

Rakendust testis töö autor manuaalse vastuvõtutestimisega. Tarkvara testija poolt läbi viidud klaviatuuri kasutatavuse testimine, millele autor tagas sisendi, saadi tagasisidet ka klaviatuuri navigeerimise toimimisele. Testimiste tulemusena selgus, et 27-st testimisse võetud kriteeriumist vastas 24 nõuetele ning ülejäänud kolme parandused jäävad järgmisse etappi. Tarkvara testija poolt läbiviidud üldine kasutatavuse testimine tuvastas kitsaskohti, mis lahendatakse teise etapi arenduse jooksul.

Rakenduse lähtekood on saadaval Bitbucketis, teostatud disainilahendused on Figma ning allalaetava rakenduse link koos eelnevalt mainitutelega on välja toodud peatükis 4.2. Rakendus toimib Windowsi operatsioonisüsteemil, MacOSil kui ka Linuxis.

EbA-UIM rakenduse digiligipääsetavust on võimalik edasi arendada, luues paremaid võimalusi nägemispiirangutega inimestele. Tulenevalt tehtud analüüsist on ka funktsionaalsusi, mille lisamine edendaks oluliselt analüüsimise protsessi. Nendeks funktsionaalsusteks on märkmete tegemine tabeli lahtritesse, tabeli ridade vahetamine ja ridade vahele lisamine.

Viidatud kirjandus

- [1] World Wide Web Consortium. Introduction to Web Accessibility. <https://www.w3.org/WAI/fundamentals/accessibility-intro/> (15.5.2025)
- [2] Riigikantselei. Ligipääsetavuse rakkerühma lõpparuanne. 2021. https://www.riigikantselei.ee/sites/default/files/documents/2021-09/Riigikantselei_Ligip%C3%A4%C3%A4setavuse_rakker%C3%BChma_%20lopparuanne.pdf (15.5.2025)
- [3] - Piirisild A., Perandrés Gómez A., Taveter K. A New Usability Inspection Method: Experience-Based Analysis. *Requirements Engineering: Foundation for Software Quality*. Cham: Springer Nature Switzerland, 2024, 74–91. https://doi.org/10.1007/978-3-031-57327-9_5
- [4] Piirisild A. Tool for the Experience-based Analysis (EbA) method. 2024. <https://doi.org/10.23673/re-453>
- [5] Piirisild A. Analoogetel põhinev tehnoloogia tulemuslikkuse ennustumudel ja töövahend. TÜ ühiskonnateaduste instituudi magistritöö. 2021. <http://hdl.handle.net/10062/72631>
- [6] Kreinin. I. Kasutajaliidese disaini loomine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele. TÜ arvutiteaduse instituudi bakalaureusetöö. 2024. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=79737
- [7] Kuldmaa K. O. Rakendusprogrammi loomine „Kasutatavuse kogemuspõhise analüüsi meetodile“. TÜ arvutiteaduse instituudi bakalaureusetöö. 2024. https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=79625
- [8] Eesti Puuetega Inimeste Koda. Mida me teeme: Ligipääsetavus. <https://epikoda.ee/mida-me-teeme/ligipaasetavus> (16.4.2025)
- [9] World Health Organization. Disability – Key facts. https://www.who.int/health-topics/disability#tab=tab_1 (17.4.2025)
- [10] Díaz-Bossini J.-M., Moreno L. *Accessibility to mobile interfaces for older people. 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, DSAI 2013. ScienceDirect, 2013, 57-66.

https://www.researchgate.net/publication/258630841_Accessibility_to_Mobile_Interfaces_for_Older_People

[11] Eesti Puuetega Inimeste Koda. Statistika. <https://epikoda.ee/spetsialistile/statistika> (17.4.2025)

[12] Seixas Pereira, L., Duarte, C. Evaluating and monitoring digital accessibility: practitioners' perspectives on challenges and opportunities. *Univ Access Inf Soc* (2025). <https://doi.org/10.1007/s10209-025-01210-w> (16.4.2025)

[13] World Wide Web Consortium. Visual – Diverse abilities and barriers. <https://www.w3.org/WAI/people-use-web/abilities-barriers/visual/> (13.5.2025)

[14] Kiambati F. G., Juma S. W., Wawire B. A. Accessibility of digital systems in information retrieval by users with visual impairment. *Quality Assurance in Education*, 2024, Vol. 32, No. 4, lk 533–550. <https://doi.org/10.1108/QAE-11-2023-0190>

[15] World Wide Web Consortium. Cognitive and learning – Diverse abilities and barriers. <https://www.w3.org/WAI/people-use-web/abilities-barriers/cognitive/> (13.5.2025)

[16] World Wide Web Consortium. Diverse abilities and barriers. <https://www.w3.org/WAI/people-use-web/abilities-barriers/> (13.5.2025)

[17] Tarbijakaitse ja Tehnilise Järelevalve Amet. Digiligipääsetavuse tagamine. <https://ttja.ee/eraklient/tarbija-oigused/kaubandus-teenused/digiligipaasetavuse-tagamine#sissejuhatus-digilig> (16.4.2025)

[18] - WCAG. WCAG 101: Understanding the Web Content Accessibility Guidelines. <https://www.wcag.com/resource/what-is-wcag/> (12.12.2024)

[19] Mets M.-E. Ligipääsetavuse reeglid, mida mõistetakse tihti valesti. *TWN Blogi*. <https://blog.twn.ee/et/ligipaasetavuse-reeglid-mida-moistetakse-tihti-valesti> (12.12.2024)

[20] University of Minnesota Duluth, ITSS. Strengths and Limitations of Automated Tools. https://www.d.umn.edu/itss/training/online/wave/strengths_limits.html (15.5.2025)

[21] World Wide Web Consortium. Selecting Web Accessibility Evaluation Tools. <https://www.w3.org/WAI/test-evaluate/tools/selecting/> (15.2.2025)

[22] Web.dev. Manual accessibility testing. <https://web.dev/learn/accessibility/test-manual> (12.12.2024)

[23] - The World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.2. <https://www.w3.org/TR/WCAG22/> (12.12.2024)

[24] The World Wide Web Consortium. How to Meet WCAG (Quick Reference). <https://www.w3.org/WAI/WCAG22/quickref/#time-based-media> (12.12.2024)

[25] European Telecommunications Standards Institute (ETSI). EN 301 549 V3.2.1 (2021-03): Accessibility requirements for ICT products and services. https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf (13.5.2025)

[26] European Telecommunications Standards Institute (ETSI). EN 301 549 V3 – The harmonized European standard for ICT accessibility. <https://www.etsi.org/human-factors-accessibility/en-301-549-v3-the-harmonized-european-standard-for-ict-accessibility> (13.5.2025)

[27] European Commission. *European Accessibility Act*. https://commission.europa.eu/strategy-and-policy/policies/justice-and-fundamental-rights/disability/union-equality-strategy-rights-persons-disabilities-2021-2030/european-accessibility-act_en (13.5.2025)

[28] European Commission. Veebi juurdepääsetavuse direktiiv – standardid ja ühtlustamine. <https://digital-strategy.ec.europa.eu/et/policies/web-accessibility-directive-standards-and-harmonisation> (13.5.2025)

[29] Põldma, S. „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakenduse edasiarendus ja tiimitöö juhtimine. TÜ arvutiteaduse instituudi bakalaureusetöö. 2025.

[30] Germann, G. Testimised ja macOSile kohandamine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele. TÜ arvutiteaduse instituudi bakalaureusetöö. 2025.

[31] Angular Material. Tabs – Overview. <https://material.angular.dev/components/tabs/overview#ally> (13.5.2025)

[32] Pandit P., Tahiliani S., Sharma M. Distributed agile: Component-based user acceptance testing. 2016 Symposium on Colossal Data Analysis and Networking (CDAN), IEEE, 2016, lk 1-9. <https://doi.org/10.1109/CDAN.2016.7570913>

[33] Goldstein K. *Some experimental observations concerning the influence of colors on the function of the organism.* *Occupational Therapy and Rehabilitation*, 1942, vol. 21, pp. 147–151. <https://doi.org/10.1097/00002060-194206000-00002>

Lisad

I. WCAG 2.2 kriteeriumite valik

Lisas 1 on toodud tabeli kujul WCAG 2.2 standardi kriteeriumite valik, mis näitab, millised kriteeriumid ei kohaldu EbA-UIM rakenduse teises etapis, millised jäävad välja ning millised kuuluvad testimisele. Kõrvuti on näidatud tabelis esimese etapi testimisse valitud kriteeriumeid, mida on kõrvutatud teises arenduse etapis vaatluse alla võetud kriteeriumitega. Esimesed kolm põhjendust P1, P2 ja P3 on esimese etapi disaineri sõnastatud põhjendused [6] ja ülejäänud on käesoleva töö autori sõnastatud põhjendused.

P1: „Kriteeriumi tuleb testida rakenduses ja on seotud arendamisega. Jääb testimisest välja, kuna kriteeriumi pole võimalik testida ainult disainilahenduse põhjal ning vead ei pruugi sõltuda üksnes disainilahendusest” [6].

P2: „EbA-UIM rakendusse ei tule sellist sisu, mida kriteeriumi testimiseks vaja on” [6].

P3: Tegu on aegunud kriteeriumiga [6].

P4: Ei võetud vaatluse alla, kuna antud kriteerium on AAA tase ning eesmärk oli täita välja valitud A ja AA taseme kriteeriumid.

P5: Erandina ei kehti see kriteeriumi EbA-UIM rakenduse kohta.

P6: Selle kriteeriumi sisu on eelnevas etapis testitud, vastav sisu vastas kriteeriumile ning uut sisu selle kriteeriumi edasiseks testimiseks ei tekkinud.

P7: Kriteeriumi täitmine ja testimine jääb rakenduse arenduse järgmisesse etappi.

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
1. Tajutav (<i>Perceivable</i>)					
1.1.1	A	Jääb välja	P1	Testimisele	-
1.2.1	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
1.2.2	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
1.2.3	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
1.2.4	AA	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
1.2.5	AA	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
1.2.6 - 1.2.9	AAA	P4		Ei ole kohaldatav	P2
1.3.1	A	Jääb välja	P1	Testimisele	-
1.3.2	A	Jääb välja	P1	Testimisele	-
1.3.3	A	Testimisele	-	Testimisele	-
1.3.4	AA	Jääb välja	P1	Jääb välja	P5
1.3.5	AA	Jääb välja	P1	Ei ole kohaldatav	P2
1.3.6	AAA	P4		Testimisele	-
1.4.1	A	Testimisele	-	Testimisele	-
1.4.2	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
1.4.3	AA	Testimisele	-	Testimisele	-
1.4.4	AA	Jääb välja	P1	Jääb välja	P7
1.4.5	AA	Testimisele	-	P6	
1.4.6	AAA	P4		Jääb välja	P7
1.4.7	AAA	P4		Ei ole kohaldatav	P2
1.4.8	AAA	P4		Jääb välja	P7
1.4.9	AAA	P4		Jääb välja	P7
1.4.10	AA	Jääb välja	P1	Ei ole kohaldatav	P5

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
1.4.11	AA	Testimisele	-	Testimisele	-
1.4.12	AA	Jääb välja	P1	Jääb välja	P7
1.4.13	AA	Jääb välja	P1	Testimisele	-
2. Kasutatav (Operable)					
2.1.1	A	Jääb välja	P1	Testimisele	-
2.1.2	A	Jääb välja	P1	Testimisele	-
2.1.3	AAA	P4		Testimisele	-
2.1.4	A	Jääb välja	P1	Ei ole kohaldatav	P2
2.2.1	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
2.2.2	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
2.2.3	AAA	P4		Ei ole kohaldatav	P2
2.2.4	AAA	P4		Ei ole kohaldatav	P2
2.2.5	AAA	P4		Ei ole kohaldatav	P2
2.2.6	AAA	P4		Ei ole kohaldatav	P2
2.3.1	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
2.3.2	AAA	P4		Ei ole kohaldatav	P2
2.3.3	AAA	P4		Ei ole kohaldatav	P2
2.4.1	A	Testimisele	-	Testimisele	-
2.4.2	A	Jääb välja	P1	Jääb välja	P7

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
2.4.3	A	Jääb välja	P1	Testimisele	-
2.4.4	A	Testimisele	-	Testimisele	-
2.4.5	AA	Testimisele	-	Testimisele	-
2.4.6	AA	Testimisele	-	Testimisele	-
2.4.7	AA	Jääb välja	P1	Testimisele	-
2.4.8	AAA	P4		Jääb välja	P7
2.4.9	AAA	P4		Testimisele	-
2.4.10	AAA	P4		Jääb välja	P7
2.4.11	AA	Jääb välja	P1	Testimisele	-
2.4.12	AAA	P4		Testimisele	-
2.4.13	AAA	P4		Testimisele	-
2.5.1	A	Ei ole kohaldatav	P2	Jääb välja	P7
2.5.2	A	Jääb välja	P1	Testimisele	-
2.5.3	A	Jääb välja	P1	Jääb välja	P7
2.5.4	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
2.5.5	AAA	P4		Jääb välja	P7
2.5.6	AAA	P4		Jääb välja	P7
2.5.7	AA	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
2.5.8	AA	Testimisele	-	Testimisele	-
3. Mõistetav (<i>Understandable</i>)					
3.1.1	A	Jääb välja	P1	Testimisele	-
3.1.2	AA	Jääb välja	P1	Ei ole kohaldatav	P2
3.1.3	AAA	P4		Ei ole kohaldatav	P2

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
3.1.4	AAA	P4		Ei ole kohaldatav	P2
3.1.5	AAA	P4		Ei ole kohaldatav	P2
3.1.6	AAA	P4		Ei ole kohaldatav	P2
3.2.1	A	Jääb välja	P1	Testimisele	-
3.2.2	A	Testimisele	-	Testimisele	-
3.2.3	AA	Testimisele	-	Testimisele	-
3.2.4	AA	Testimisele	-	Testimisele	-
3.2.5	AAA	P4		Testimisele	-
3.2.6	A	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
3.3.1	A	Jääb välja	P1	Ei ole kohaldatav	P2
3.3.2	A	Testimisele	-	Testimisele	-
3.3.3	AA	Testimisele	-	Ei ole kohaldatav	P7
3.3.4	AA	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
3.3.5	AAA	P4		Testimisele	-
3.3.6	AAA	P4		Testimisele	-
3.3.7	A	Testimisele	-	Testimisele	-
3.3.8	AA	Ei ole kohaldatav	P2	Ei ole kohaldatav	P2
3.3.9	AAA	P4		Ei ole kohaldatav	P2
4. Tugev (<i>Robust</i>)					
4.1.1	A	Ei ole kohaldatav	P3	Ei ole kohaldatav	P3

WCAG-kriteerium	Tase	Otsus I etapis	I etapi otsuse põhjendus	Otsus II etapis	II etapi otsuse põhjendus
4.1.2	A	Jääb välja	P1	Testimisele	-
4.1.3	AA	Jääb välja	P1	Testimisele	-

II. Ülesannete jaotus eessüsteemi arendajate vahel

Tabel kujutab käesoleva töö autori ülesannete voogu koostöös nõuete analüütikuga analüüsist kuni arenduseni. T tähistab nõuete analüütikut ja R tähistab töö autorit. Veergude „Analüüs” ja „Arendus” lahtrites olevad numbrid tähistavad ülesannete täitmise järjekorda vastavalt, kes on tegija ning mida tehti.

		Analüüs		Arendus	
ID	Ülesanne	T	R	T	R
Kõrge prioriteet (<i>must-to-have</i>)					
J4	Nelik-vastusevariandile viienda variandi lisamine.	1			2
J5	Tabeli täies ulatuses kuvamine		1		2
F1	Viimase tulba taustavärvide muutmise vastavalt tähendusele		1		2
A4	WCAG uusima standardi kohandamisega seotud muudatused		1		2
A6	Projektis ülemiste sektsioonide paigutus	1	2		3
Keskmise prioriteet (<i>should-have</i>)					
F8	Lisada “üles” nupp		1		2
F9	Mitme näite prioritseerimisvõimaluse kuvamine	1, 3	2		4
F10	Mitme näite prioritseerimise tulemuse kuvamine	1, 3	2		4
F11	Liidetud lause sees sidusrühma nimetus boldis				1
F12	Üheks lauseks liidetud tulba peitmine	1	2		3
A10	Nupu “Lisa rida” asukoht		1		
A11	Veateadete hoiatussildid		1		2
Madal prioriteet (<i>nice-to-have</i>)					
F15	Nelikvastuse muutmisel järelduse juurde NB-teavitust		1		2

		Analüüs		Arendus	
ID	Ülesanne	T	R	T	R
	Lisandunud nõuded (<i>extra</i>)				
F28	Küsimuse osa eraldamine				1
F30	Ekraanilugeri võimaldamine		1		2

III. Klaviatuuri kasutatavuse testimise sisend

Järgnevalt on välja toodud ülesannete voog, mida klaviatuuri kasutatavuse testija läbis. Ülesannetes, mille täitmisel pidi testija sisestama ettenähtud väljale teksti, on tekstisisestuse kohad märgitud jutumärkidesse pandud kolme punktiga. Tekst sisestamiseks oli ettevalmistatud testi läbiviija poolt [30].

Ülesanded:

- Eeltingimus: rakendus on käivitatud
- 1. Liikuda lehele “Meist”
- 2. Kerida lehel alla, et tuleks nähtavale “Üles” nupp ning kasutada seda tagasi algusesse liikumiseks
- 3. Liikuda lehele “EbA meetodi juhend”
- 4. Leida üles info kombineeritud vastuste tõlgendamise kohta
- 5. Leida üles info EbA meetodi 3. etapi kohta
- 6. Liikuda tagasi menüüribale
- 7. Vahetada keel teiseks ja seejärel tagasi, mis oli alguses
- 8. Liikuda lehele “Projektid”
- 9. Luua uus projekt nimega “Test Projekt”
- 10. Lisada uus sidusrühm nimega “...”
- 11. Peita tabeli kohal asetsevad paneelid
- 12. Lisada ID “T1” esimeses reas asuval funktsioonile
- 13. Anda funktsioonile kirjeldus “...”
- 14. Määrata sidusrühmaks eelnevalt loodud uus sidusrühm
- 15. Anda funktsiooni eeltingimusele kirjeldus “...”
- 16. Anda võrreldava situatsiooni kirjeldus “...”
- 17. Määrata neljale küsimusele vastused vastavalt “...”, “...”, “...” ja “...”
- 18. Peita teine (“Kirjeldus, mil määral on eeltingimus täidetud”) genereeritud lause tulp
- 19. Kategoriseeri sama rea tegevuskava tulp valitud värviga
- 20. Lisa olemasolevale funktsiooni eeltingimusele teine võrreldav situatsioon “...”
- 21. Määrata uue situatsiooni neljale küsimusele vastused vastavalt “...”, “...”, “...” ja “...”
- 22. Prioritiseerida teine situatsioon
- 23. Lisada uus rida
- 24. Kustutada lisatud uus rida

Klaviatuuri kasutatavuse testimisel kasutati ülesannete täitmise hindamiseks ja probleemide tuvastamiseks all olevat tabeli formaati. Tabeli formaadi on koostanud antud bakalaureusetöö autor.

Pealkiri	
Kasutaja	
Soov	
Eesmärk	
Eeltingimused	
Eeldatavad sammud	
Oodatav tulemus	
Tulemus/ probleemid	

IV. Testitud nõuded

Tabel annab ülevaate, millised WCAG kriteeriumid kehtivad millise nõude kohta. Tabeli esimeses veerus on kõik vaatluse alla võetud WCAG kriteeriumid, mis on kajastatud ka lisas 1. Tulba päises on teises etapi nõuded ID tähistega. Lahtrisse on märgitud “x”, kui selle rea kriteerium kehtib selle veeru lahenduse kohta.

WCAG	Teise etapi nõuded (ID)											
	J4	J5	A6	F1	F8	F9	F10	F11	F12	F15	F28	F30
1.1.1												x
1.3.1	x	x				x		x	x			x
1.3.2	x	x						x				
1.3.3				x								
1.3.6	x			x	x	x		x		x		x
1.4.1	x			x			x	x			x	
1.4.3				x			x					
1.4.11	x			x			x				x	
1.4.13				x								
2.1.1	x	x		x	x	x			x			x
2.1.2	x	x	x	x	x	x	x		x	x	x	x
2.1.3	x	x	x	x	x	x	x		x	x		x
2.4.1					x							
2.4.3	x	x			x	x			x			
2.4.4												x
2.4.5												
2.4.6												x
2.4.7		x		x	x							
2.4.9												x
2.4.11							x					
2.4.12							x					

	Teise etapi nõuded (ID)											
WCAG	J4	J5	A6	F1	F8	F9	F10	F11	F12	F15	F28	F30
2.4.13	x	x		x	x	x			x	x		
2.5.2	x	x		x	x	x						
2.5.8					x							
3.1.1												x
3.2.1	x	x			x				x			
3.2.2	x									x		
3.2.3			x								x	
3.2.4						x	x			x		
3.2.5		x							x	x		
3.3.2						x				x		
3.3.5												
3.3.6												
3.3.7										x		
4.1.2	x	x				x			x	x		x
EN 301 549	J4	J5	A6	F1	F8	F9	F10	F11	F12	F15	F28	F30
EN 5.2												x
EN 11.6.1												x
EN 11.6.2												x
EN 12.1.1												x

V. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Rahel Pettai,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Kasutajaliidese disaini ja digiligipääsetavuse edasiarendus „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele“, mille juhendaja on Anu Piirisild, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Rahel Pettai

15.05.2025